EFFICIENT MULTI-RESOLUTION DATA DISSEMINATION IN WIRELESS SENSOR NETWORKS

A Dissertation

by

JIAN CHEN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2005

Major Subject: Computer Science

EFFICIENT MULTI-RESOLUTION DATA DISSEMINATION IN WIRELESS SENSOR NETWORKS

A Dissertation

by

JIAN CHEN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Udo Pooch
Committee Members,	Frank M. Shipman
	Riccardo Bettati
	Michael Longnecker
Head of Department,	Valerie E. Taylor

August 2005

Major Subject: Computer Science

ABSTRACT

Efficient Multi-resolution Data Dissemination
in Wireless Sensor Networks. (August 2005)
Jian Chen, B.E., Qingdao University, PR China;
M.S., Peking University, PR China
Chair of Advisory Committee: Dr. Udo Pooch

A large-scale distributed wireless sensor network is composed of a large collection of small low-power, unattended sensing devices equipped with limited memory, processors, and short-range wireless communication. The network is capable of controlling and monitoring ambient conditions, such as temperature, movement, sound, light and others, and thus enable smart environments. Energy efficient data dissemination is one of the fundamental services in large-scale wireless sensor networks. Based on the study of the data dissemination problem, we propose two efficient data dissemination schemes for two categories of applications in large-scale wireless sensor networks. In addition, our schemes provide spatial-based multi-resolution data dissemination for some applications to achieve further energy efficiency. Analysis and simulation results are given to show the performance of our schemes in comparison with current techniques.

To my wife, Li Xiao, and my parents.

ACKNOWLEDGMENTS

I wish to thank my advisor, Dr. Pooch, for his help in the past several years. I am honored to be his student, and I am grateful for his kindness and help.

I wish to thank Dr. Bettati, Dr. Shipman and Dr. Longnecker for being on my committee.

I also wish to thank Mr. Willis Marti for recommending I become Dr. Pooch's student.

I want to thank Dr. Friesen for his help during the entire period I studied at Texas A&M University, especially when Dr. Pooch took medical leave.

I want to thank Dr. Yong Guan for spending time discussing research with me.

Finally, I would like to thank my family for their support and love in my endeavors. In particular, I want to express my gratitude to my wife. Together we went through hard times and conquered many difficulties making the memory valuable and beautiful.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE | Page

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Large-scale distributed sensor networks, comprised of a large collection of small low-power, unattended sensing devices equipped with memory, processors, and short-range wireless communication, can enable "smart environments" which is capable of controlling and monitoring ambient conditions such as temperature, movement, sound, light, location and others [12].

As one of the fundamental services in wireless sensor networks, energy efficient data-gathering techniques are required to extract relevant sensed data from or within a sensor network. Based on where to store the generated data, data gathering techniques are divided into three categories: local-storage, external-storage and in-network data-centric storage [39]. Early local-storage based methods such as Directed Diffusion (DD) [25], Declarative Routing Protocol [10] are not efficient in that queries are sent from sinks using flooding and, in consequence, sinks have to keep broadcasting queries periodically to maintain paths for data retrieval from data sources. Data-centric storage techniques eliminate query flooding by storing sensed data at storage points, which may be different from the location where the data is generated, using a Geographic Hashing Table [39]. The sink sends queries directly to storage points when it wants to retrieve data from the network. However, when the volume of sensed data to be sent to storage points is large, data-centric storage is not efficient for the energy consumption in this process [39].

Multi-resolution data is important to achieve energy efficiency for many sensor applications in which data of less detail is acceptable under some circumstance. Just

---

The journal model is *IEEE Transactions on Computers*.

like taking pictures, sometimes a picture with less resolution is preferred under certain conditions, although mostly we like a picture as sharp as possible.

Decreasing transmission of redundant data is also important for energy efficiency. Data redundancy is inevitable considering the unattended style of sensor deployment. In many cases, data on some nodes is a subset of data from neighboring nodes. Since the transmission of data is the most energy expensive operation for wireless nodes, minimizing the transmission of redundant data would significantly improve the performance in energy efficiency.

To meet these requirements, this study aims at addressing efficient multi-resolution data dissemination problem in large-scale wireless sensor networks. Specifically, this research covers the following research subjects. A spatial-based, application oriented multi-resolution model for data dissemination in sensor networks is proposed, based on which we propose following two efficient data dissemination schemes for two different categories of applications respectively to retrieve multi-resolution data from or within sensor networks.

- The first method is for applications like temperature monitoring, in which sensor networks are deployed to monitor a certain field and queries are sent from the sink to nodes requesting interested data periodically. In addition to saving energy by eliminating query flooding and extra data transmission for data storage elsewhere, this method achieves further energy efficiency by reducing dissemination of redundant data. In this method, instead of being addressed to nodes, queries are sent to pixel points, a set of carefully selected locations across the sensor field, on the support of a geographic routing protocol. The query finally reaches the home node of the pixel point, the node geographically closest to the pixel point, which reports data to sinks in response. The density of the pixel points corresponds to data resolution level. The higher the density, the higher the data resolution level [5].

- The second method is for applications like habitat monitoring, in which sensor networks are deployed to detect some mobile objects such as birds or animals. In this scheme, generated data is stored locally while data sources register to registration points (geographical locations dispersed across the network) using data-centric storage techniques. When sinks solicit data from the network, queries with resolution specifications are sent to and stored at all registration points which forward them to all matched data sources. When a data source receives a query, a set of nodes corresponding to the resolution specification in the query is selected to report data to sinks. The nodes with data of high redundancy are prevented from being queried and thus the dissemination of redundant data is reduced [8].

Analysis and simulation results show that the performance of our methods are much better than early flooding-based methods especially in large-scale networks with relatively high node density.

The rest of the dissertation is organized as follows. Chapter II is literature review. In Chapter III and Chapter IV, two data dissemination schemes for two categories of applications are presented. Chapter V presents a fundamental geographic routing protocol to support the data dissemination schemes. Conclusion and future works are given in Chapter VI.

CHAPTER II

LITERATURE REVIEW

A. Some Unique Properties of Wireless Sensor Networks

Sensor networks have some unique properties which distinguish themselves from other wireless networks. The following characteristics of sensor networks are particularly considered when we design our multi-resolution data dissemination schemes.

- Sensor nodes have stringently limited resources such as memory, CPU, and, in particular, power resource. Considering the potential applications for wireless sensor networks like deployment in the battle field to detect enemies, refilling of energy is unreachable or unreasonable. As a result, energy efficiency is always the primary goal of any protocols and algorithms for wireless sensor networks.

- Because energy is an extremely stringent resource in sensor network performing local computation to reduce data before transmission can obtain orders of magnitude energy saving [35]. Hence, algorithms in sensor networks always try to reduce the communication to the minimum.

- Sensor networks may be deployed in a large scale, composed of a large number of small and cheap sensor nodes capable of wireless communication and significant computation. The large number of nodes is to make up the unattended manner of deployment and the incapability of individual node. Traditional network technologies for the Internet cannot be directly applied to sensor networks. For example, a sensor node may not need an identity [12]. Spatial location plays a more and more important role in sensor networks [22].

- Sensor networks may be deployed in a very ad hoc manner (such as physical installation of each sensor or random aerial scattering from an airplane) and must automatically adapt to changes in environments [14]. There is lack of *a-priori* knowledge of *post-deployment* configuration. Therefore, distributed and localized algorithms are preferred in sensor networks in that it is more energy efficient by reducing communication, and robust to network topology change [12].

- Dynamic nature is an intrinsic property of sensor networks even under the assumption of static (i.e. immobile) nodes in this paper. Sensor nodes are devices usually small in the dimension, energy stringent and less reliable easily falling into dysfunction state or dead due to energy exhaustion or hardware failure. New nodes may be replenished periodically. Furthermore,they are apt to be affected by various factors like environment. For example, nodes may be intermittently unreachable as a result of impact of moving objects in the monitored environment.

B. Data Dissemination in Wireless Sensor Networks

Data centric approaches are now seen to be fundamental to sensor networks [3], [23], [25], [33]. As stated in GHT [39], based on where the events generated by processing of lower-level time series sensed data are stored, data centric approaches on data gathering problem can be divided into three categories: local-storage, in-network data centric storage and external-storage. Under the assumption that events occur at locations not known in advance and therefore queries are flooded to all nodes, local storage is the preferable from the performance standpoint when the number of generated events is large compared to the system size and all events are retrieved rather than a summary. Data centric storage is preferred when the sensor network is large in comparison to the event number and only a summary of a large volume of events are queried.

Directed Diffusion [25] and Declarative Routing Protocol [10] can be categorized into local-storage methods and do not require any routing techniques besides flooding. Directed Diffusion and Declarative Routing Protocol both take the data-centric naming approach to enable in-network data aggregation. Directed Diffusion initiates low-rate data flooding and gradually reinforce better paths to accommodate certain levels of network and sink dynamics. Declarative Routing Protocol uses restricted flooding to improve the performance.

GHT proposes the data-centric storage technique first. There are substantial work with respect to data centric storage approach such as [16], [19], [31], [39], which require the support of geographic routing techniques.

SPIN [24] considers application-specific knowledge during data dissemination among early efficient data dissemination methods. What it considers is its *meta-data*'s application-specific knowledge and knowledge of resource available before making its communication decisions.

TTDD [45] is an approach for data dissemination in the scenario of mobile sinks. In TTDD, mobile sinks are able to continuously receive data on the move by flooding queries within a local cell which is among a grid structure proactively built.

Some applications provide multi-resolution services in sensor networks. DIFS [19] is an extension of GHT to support efficient range queries while maintaining balanced load balance across nodes. It achieves this by the construction of a multiple rooted hierarchical index in which non-root nodes can have multiple parents. As GHT, the events stored in the index are high-level events which are abstractions of the lower-level time series data. Multi-dimensional Range Queries [31] designs a distributed index for multi-dimensional data that scalably supports multi-dimensional range queries. Events with comparable attribute values are stored nearby by using a locality-preserving geographic hash, which maps the multi-dimensional space(described by the set of attributes) to two dimension geographic space.

DIMENSIONS [16] takes advantage of both data-centric storage and local-storage techniques. It constructs multi-resolution summaries of sensor data and stores them in the network in a spatially and hierarchically decomposed distributed storage structure which is optimized for efficient searching. However, it has disadvantages of data-centric storage techniques. Extra energy is consumed for communication between nodes and data-centric storage points. In addition, the nodes with redundant data still send summaries to data storage points, which expends valuable energy resource. Furthermore, even sinks have no tendency to retrieve data, the hierarchical summaries are still constructed and maintained, which incurs energy inefficiency for some applications.

A sensor node with the radio off could save energy in orders of magnitude than the state with the radio on. Topology control techniques are trying to maximize the number of nodes and the length of time they go to sleep thereby achieving energy efficiency to extend the lifetime of the entire network. Some of the most well known representatives of topology techniques in sensor network are proposed in [4], [9], [38], [41], [44]. Our method is able to accommodate these techniques well.

Connected Sensor Cover [20] considers the sensing region of sensors and designs algorithms for selecting a minimum subset of connected sensor nodes whose sensing regions cover the queried area.

Coping irregularity [15] points out that spatial-temporal irregularity is fundamental to wireless sensor networks. It lists several cases for the existence of such irregular spatial-temporal sampling and the impact on many performance issues in sensor networks.

C. Geographic Routing Protocols in Wireless Networks

There have been many geographic routing algorithms in wireless networks such as [2], [17], [26], [27], [30], [32], [42], [46], and others. Nevertheless, most existing geographic routing protocols are not customized for sensor networks and have to be modified to adapt to the unique properties of sensor networks before being applied in sensor network applications.

GPSR [26] is one of the most well known geographic routing algorithms in wireless networks. Applications in [16], [19], [31], [39] are all built atop GPSR in sensor networks. In the implementation of data centric storage in GHT, GPSR is adopted and modified to fit the properties of the sensor network. First, packet destination is marked with locations instead of identifiers like node IP address. Second, the home node for the target location is identified by the *home perimeter traversal* method. *Home perimeter* is composed of all nodes surrounding a location in which home node is the nearest to the location. After taking a tour of the enclosed home perimeter of the target location and returning to the nearest node to the target location, a packet notices the loop and recognizes it as the home node of the target location. Third, to solve the data consistency problem, the home node recruits all nodes on the home perimeter as replica nodes. Fourth, it points out the boundary problem but does not give solutions.

The impact of radio irregularity on wireless sensor network including geographic routing protocols is studied in [28], [43], [47]. The impact of non-uniform transmission range on GPSR is pointed out in [13]. Methods to guarantee successful perimeter routing for those geographic routing methods based on planar graph by adding virtual edges in cases of instable transmission ranges are proposed in [1]. This result is extended towards efficiency in [29].

CHAPTER III

A SPATIAL-BASED MULTI-RESOLUTION DATA DISSEMINATION SCHEME

A. Introduction

In order to better understand the problem, we begin with a brief study of the properties of applications this scheme is applied to. A good example of this category of applications is temperature monitoring which has following properties.

First, sensor networks are deployed to monitor a certain field and queries are sent from the sink to nodes requesting interested data periodically. It is always the sink which initiates the data retrieval and matched data is to be sent to the sink from data sources upon receiving queries.

Second, data redundancy is inevitable for a large-scale wireless sensor network with relatively high density of nodes. In sensor networks, a diversity of sensors with different sensing ranges are employed for various tasks. The sensor embedded in the node covers a certain area within its sensing range in which there may exist other sensor nodes. For a task fulfilled by sensors with large sensing range in a sensor network with high node density, retrieving data from every node is not only unnecessary because some data is duplicated, but also waste of energy caused by transmission of redundant data. Decreasing transmission of redundant data would significantly improve the performance in energy efficiency. However, efficiently distinguishing sensor nodes with redundant data from others and having them not queried is a challenge.

Third, less detailed data is acceptable for such applications under some circumstance for the purpose of energy efficiency. For example, the temperature data retrieved from a small subset of nodes, which are uniformly selected from the sensor field based on their geographic locations, could be accepted as an approximation of the condition across the network. Just like taking pictures, sometimes we like a picture as sharp as possible, whereas sometimes a picture with less resolution is acceptable. By sending less data in lower resolution level across the networks, energy expenditure on communication could be decreased significantly. Therefore, multi-resolution data dissemination is important for such applications from the energy efficiency standpoint to provide data retrieval in multiple resolution levels as requested. In image processing, multi-resolution model that represents objects at multiple levels of detail are required for this purpose [21]. Similarly, in sensor applications, an algorithm that automatically selectively retrieves subsets of the data in correspondence to the detail requirement, while preserving high fidelity, is essential to provide multiple resolution data dissemination.

Fourth, sometimes, users are interested in the conditions of certain sub-areas of the sensor field with various resolution requirements. In other words, users may want to retrieve high detailed data from a certain area of the sensor field and request data of less resolution from another area at the same time. A data dissemination scheme which could provide such flexible services is preferred.

To take advantage of the properties of the potential applications and meet the above requirements, we propose a spatial-based multi-resolution data dissemination scheme. In addition to saving energy by eliminating query flooding and extra data transmission for data storage elsewhere, this scheme accomplishes further energy efficiency by reducing dissemination of redundant data. Moreover, this scheme provide efficient, flexible multi-resolution data dissemination services so that users are able to retrieve data at various resolution levels from different areas at the same time. Note that the data we discuss in this dissertation is the high-level events, rather than the lower-level time series data from which events are composed [19].

In this scheme, a sensor field is divided into many pieces of small regions (called

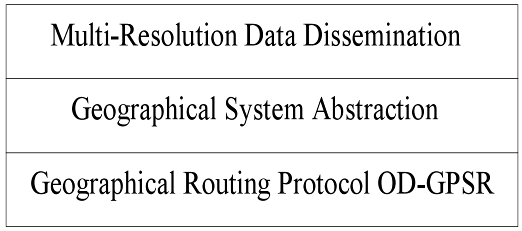

Fig. 1. The architecture of the multi-resolution data dissemination scheme in wireless sensor networks

grid). The grid size corresponds to the data resolution level. The higher the data resolution level, the less the grid size. To retrieve data, queries are not sent directly to sensor nodes, but to locations (called *pixel points*) instead, with one pixel point inside each grid, along paths in a tree composed of the *sink* and all associated pixel points. A query reaches the *home node* of its target pixel point, the sensor node geographically closest to that location. In response, the home node reports data to its parent pixel point along the same path of the query in reverse direction, where in-network aggregation may be conducted to achieve further energy efficiency, until the data reaches the sink where the query is originated. The minimum grid size (corresponding to the highest resolution level) for a task is determined by the sensing range of sensors for the task and is small enough that the sensing ranges of pixel points in all grids cover the whole sensor field. Therefore the data retrieved from these pixel points is as detailed as that of *all-node-querying* methods in which all nodes are queried. By retrieving data from the pixel point only, redundant data on other nodes is prevented from being transmitted and valuable energy is saved without loss of fidelity.

This scheme is based on a three-level architecture shown in Fig. 1. The upper layer is *multi-resolution data dissemination* which aims at providing data of multi-level of detail about sensing environments for applications. In the middle, we have a *geo-*

graphical system abstraction layer which maps the resolution level required by a query into a set of carefully-selected sampling locations. The lower layer is a geographical routing protocol which routes packets to particular locations. To support the delivery of packets to a particular location, we use *On-demand GPSR* (OD-GPSR), a modified version of Greedy Perimeter Stateless Routing(GPSR) [26]. OD-GPSR is customized for sensor networks with several particular properties. First, it achieves energy efficiency by prohibiting unnecessary communications among neighbors. Second, OD-GPSR identifies packet destination with locations instead of node identifiers like IP address.

B. Methodology

We assume a large scale sensor network composed of *static* (i.e., immobile) energy-constrained sensor nodes with uniform transmission ranges and the density of nodes in the network is relatively high. This assumption is reasonable because, in many potential applications, sensor networks are deployed in an unattended manner due to the limited access to the monitored area. Usually a large number of sensor nodes are deployed to make up the unattended deployment and improve the coverage over the monitored area. The boundary of the monitored are is known approximately.

Energy is an extremely stringent resource in sensor networks because sensor nodes may be untethered and the replenishment of energy may be unreachable or unreasonable. Sensor nodes communicate with neighbors via short-range radio and thus long distance communication has to be passed through multiple hops. Sensor networks are *task-specific* in that the task types are known at the time the sensor network is deployed. Therefore, every node is aware of its missions.

To simplify the problem, we assume a two dimension flat area covered by sensor networks, although our method also applies to three dimension area, in which an approximation of the boundary of the network is known. Each node knows its location via GPS or other mechanisms such as [3], [18], [36], [37], [40]. The data in our scheme is *named data* which is high-level events generated by sensors rather than the lower-level time series data from which events are composed [19]. We assume that the high-level events are generated after processing the lower-level time series data by the sensor node locally either independently or coordinately with other adjacent nodes upon detection of phenomena. Therefore, even the events associated with a node are not stored in the node, they must be at adjacent nodes and can be accessed quickly by the node.

1. Application-oriented Spatial-based Multi-resolution Data Model

We assume that the area a sensor node can monitor is a circle with itself as the center. We use a simple sensor detection model in which each sensor can detect a target with some confidence if the distance of the target is within the sensing range d and fails to detect beyond that distance. For the convenience of analysis, terrain effect on the detection is not taken into account.

a. Definition of Data Resolution Levels

In this model, if the entire sensing area of a node is covered by other neighbor nodes, the data in this node is defined as redundant data because it totally overlaps with that in neighbor nodes. Ideally the highest resolution data without loss of fidelity but with no redundancy is from a minimum subset of nodes in a data source whose sensing ranges cover the whole data source area and no node's sensing area is totally covered by other nodes in the same subset. An example is shown in Fig. 2. Such a set of nodes are selected based on an important parameter, the sensing range of the

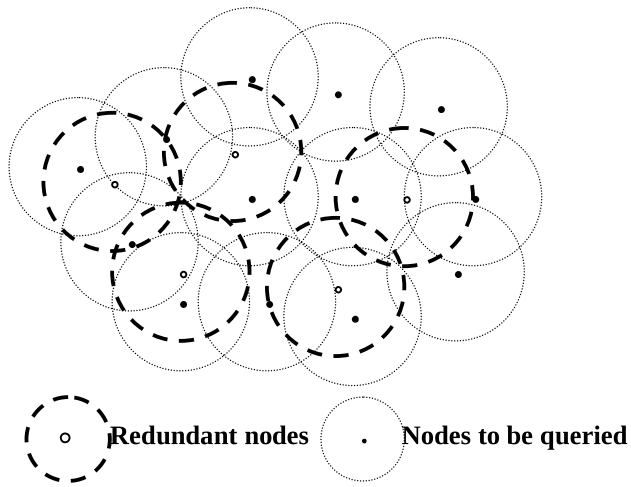

Fig. 2. An example of data in the highest resolution level

sensor for a task, and the location information of all nodes inside the data source. This is the definition of data in the highest resolution level.

For less resolution levels, data is retrieved from less number of nodes dispersed across the data source region. Data from these nodes is a less detailed approximation of the condition in that region. Such a subset of nodes for a lower resolution level of data is selected based on spatial information of nodes inside the data source and a virtual sensing range. A certain value of this virtual sensing range corresponds to a resolution level and is determined by the application. The higher the resolution level, the less the value. The least value is the actual sensing range of a sensor for a task which corresponds to the highest resolution level of data.

b. Relations Among Multiple Data Resolution Levels

Fig. 3 shows a comparison among data dissemination in multiple resolution levels in terms of energy efficiency and detail level. The data of the highest resolution level is in the same detail level as that of all-nodes-querying methods like [25] but usually with less redundancy. Data retrieval in less resolution levels is more energy efficient

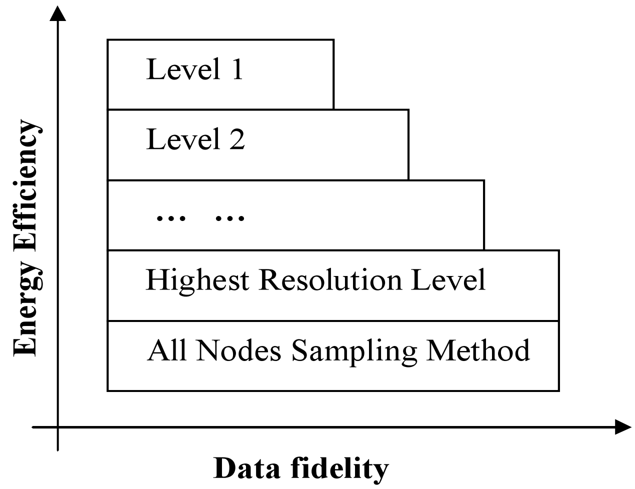

Fig. 3. The relation among multiple resolution levels

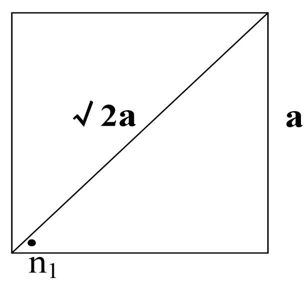

Fig. 4. Grid size for the highest resolution level

than higher resolution levels for the transmission of less data whereas at the cost of losing more fidelity of the data. The number of resolution levels for an application depends on the application. For example, an application like temperature monitoring in an outdoor large area may accept more resolution levels than an animal monitoring application, since a less detailed data is acceptable to represent temperature variations while more detailed data is required to prevent from missing animals passing.

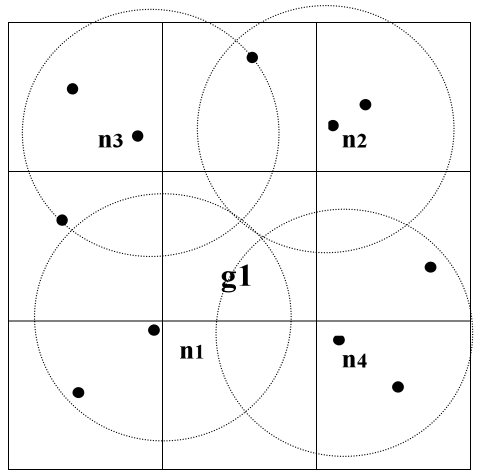

Fig. 5. Covering problem

2. Implementation of Data Retrieval in Multiple Resolution Levels

In this scheme, the sensor field is divided into grids inside each of which a *pixel point* is picked to be queried and the home node of that pixel point reports data upon receiving queries. Note that a pixel point is a geographical location. To retrieve data as detailed as that of all-node-querying method, the grid size of the highest resolution level should be small enough that the sensing ranges of home nodes of all queried pixel points cover the entire sensor field. For this reason, the length of the side of a grid for the highest resolution level is set as $a = d/\sqrt{2}$. With this value, the sensing range of any node queried inside the grid guarantees the coverage over the grid, as shown in Fig. 4. When there is no node inside a grid, an outside node closest to the pixel point geographically will be queried. Note that different tasks may use sensors with different sensing ranges and, as a result, the size of the grids for the highest resolution level for tasks may be different.

This implementation, however, has the *covering problem* for occasions as shown in Fig. 5. In this case, although there is no node inside the grid g_1, the grid is covered by four nodes n_1, n_2, n_3 and n_4 which are located inside neighbor grids. Suppose the

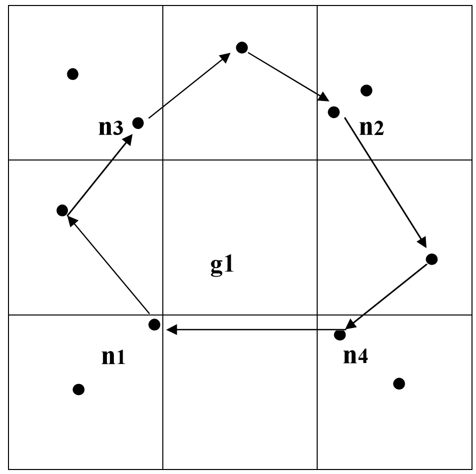

Fig. 6. Solution to the covering problem

pixel point is set at the center of the grid, n_1 will be queried since it is the closest node to grid g_1, while its knowledge of grid g_1 is not complete because it covers only about half of grid and other three nodes holding data of the rest of the grid are not queried. To solve this problem, when a home node of a pixel point in a grid finds itself not inside the grid after receiving a query, it transfers the query to all nodes on the home perimeter of the pixel point using *home perimeter traversal* algorithm, which is also used for identifying the home node for a target location in OD-GPSR. In response, these nodes report data to the parent pixel point if they have sensed data from that grid. As a result, all nodes holding data of the grid are queried, as shown in Fig. 6.

Lower resolution levels have larger grid sizes. The larger the grid size, the less the resolution level. It is obvious that, for a task using sensors with certain sensing ranges, the proportion of the grid area covered by the queried node is less for a larger grid. Fig. 7 shows an example with two resolution levels. The upper one is the grids of larger size for data retrieval in a lower resolution level, while the lower one is that of less size for data retrieval in a higher resolution level.

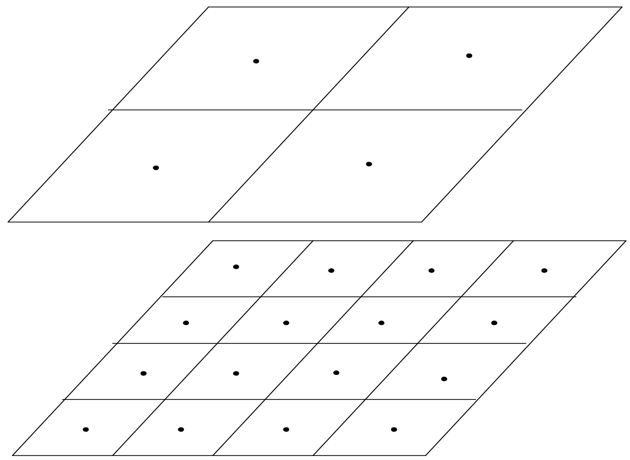

Fig. 7. Multiple resolution levels

Since the number of resolution levels is determined by the application, the mapping between a resolution level and its grid size is also determined by applications except for the highest resolution level which is determined by the real sensing range of the sensor for a task. Users can determine the number of data resolution levels as well as the grid size for each resolution level based on the individual properties of applications.

3. Querying Schemes

We present two querying schemes in this section, data-independent querying scheme and data-sensitive querying scheme.

a. Data-independent Querying Scheme

In the highest resolution level, the set of pixel points one of which is from each grid is able to cover the entire sensor field. Data extracted from all these pixel points is as detailed as that of all-node-querying method. Therefore, in this scheme, the same set of pixel points are queried every time in the highest resolution level.

For data retrieval in lower resolution levels, a disjoint subset of the pixel points of the highest resolution level are queried each time in a *cycle*, with each pixel point inside a grid of the lower resolution level(one grid in the lower resolution level may be composed of more than one grids of the highest resolution level). After a cycle, all pixel points of the highest resolution level are queried once. The purpose of this design is to guarantee all data retrieved in a cycle for lower resolution levels combined provides a piece of data as detailed as the highest resolution level and reflects the situation of the entire sensor field, although the data retrieved each time in a cycle is an less detailed approximation of data of the whole sensor field. The value of the cycle is determined by resolution levels. Note that the grid size of a lower resolution level is larger than that of the highest resolution level and therefore one grid has more than one pixel point of the highest resolution level.

Following is a formal definition of this querying scheme. We use set L to denote all pixel points queried in the highest resolution level, with one pixel point for each grid. For a certain resolution level $level_i$, in jth period p_j of a cycle (there is only one period in a cycle for the highest resolution level), a different pixel point inside each of its grids is queried, which is an element of L. Therefore a subset of L, L_j, is queried in jth period. We use T to denote the value of a cycle, the time to query all elements of L once, then,

$$L = \sum_{j=1}^{k} L_j, \;\; T = \sum_{j=1}^{k} p_j \tag{3.1}$$

while $L_m \wedge L_n = \phi$, $\forall (m \neq n)(1 \leq m, n \leq k)$ (k is the number of periods in a cycle for $level_i$)

In the following we give an implementation of this scheme. We use N_i to denote the number of grids in $level_i$, and, $N_i = 4^i$. A grid of $level_i$ is composed of 4 grids of

$level_{i+1}$. Suppose the frequency to query the network is the same and one pixel point in each grid is queried each time, the time of a cycle for $level_{i+1}$ is less than $level_i$, because the grid number of resolution $level_{i+1}$ is 4 times of $level_i$ and correspondingly the number of pixel points queried is 4 times in each period.

$$T_{i+1} = \frac{T_i}{4} \tag{3.2}$$

Algorithm 1 is used to number the grids of resolution levels in an application. In the highest resolution level, the sequence number of the grids is the same as that of the pixel points.[1]

num_grids(l, h, lx, rx, ly, ry) {

if(l==h) {
grids[lx, ly]=l;
return;
}

num_grids($l, l + \frac{h-l+1}{4} - 1, lx, lx + \frac{rx-lx+1}{2} - 1, ly, ly + \frac{ry-ly+1}{2} - 1$);
num_grids($\frac{h-l+1}{4} + l, l + \frac{h-l+1}{2} - 1, lx + \frac{rx-lx+1}{2}, rx, ly, ly + \frac{ry-ly+1}{2} - 1$);
num_grids($\frac{h-l+1}{2} + l, l + \frac{3(h-l+1)}{4} - 1, lx + \frac{rx-lx+1}{2}, rx, ly + \frac{ry-ly+1}{2}, ry$);
num_grids($\frac{3(h-l+1)}{4} + l, h, lx, lx + \frac{rx-lx+1}{2} - 1, ly + \frac{ry-ly+1}{2}, ry$);
}

Algorithm 1: Number the grids for a resolution level

Recursive Algorithm 2 is used to pick the pixel points each time of querying in a cycle for a resolution level.[2]

---

[1] Array grids[lx..rx,ly..ry] is used to keep the sequence number of the grids. $1 \leq lx, rx, ly, ry \leq 2^{hl}$; $1 \leq l, h \leq 4^{hl}$ (hl denotes the highest resolution level)

[2] Parameter *level* denotes the resolution level; *hlevel* is the highest resolution level for an application; *num_querying* means the nth time to query pixel points in the resolution level; *grid_num* is the gth grid for a resolution level; *queryingPeriod* means the mth time to query pixel points in current cycle of the resolution level.

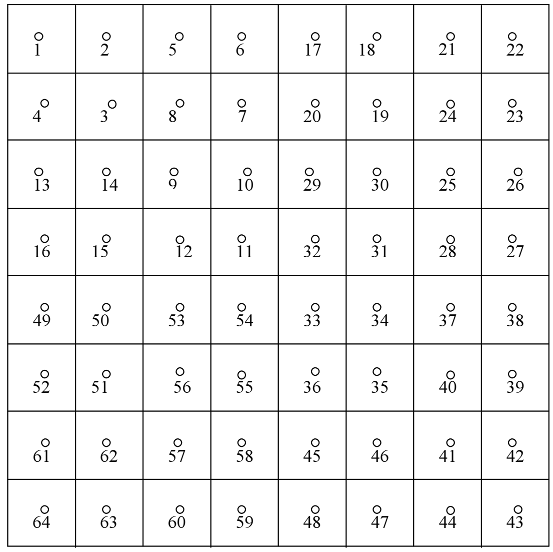

Fig. 8. All grids and pixel points for the highest resolution $level_3$ in the example

Pick-pixel-points($level, hlevel, num\_querying$) {

  if ($level > hlevel$)
    return false;

  for ($i = 1$; $i \leq 4^{level}$,$i++$)
    pick-a-pixel-point($level, hlevel, i, num\_querying \ mod \ 4^{hlevel-level}$);

}

int pick-a-pixel-point($level, hlevel, grid\_num, queryingPeriod$){

  if (level==hlevel)
    return $grid\_num$;
  else
    return pick-a-pixel-point($level + 1, hlevel, 1 + 4 * (grid\_num - 1)+$
      $(queryingPeriod - 1) \ mod \ 4$, $\lceil \frac{queryingPeriod}{4} \rceil$);

}

Algorithm 2: Pick pixel points for querying for a resolution level

We use an example to show how this querying scheme works. In this example we assume an application with data dissemination in two resolution levels, $level_2$ and

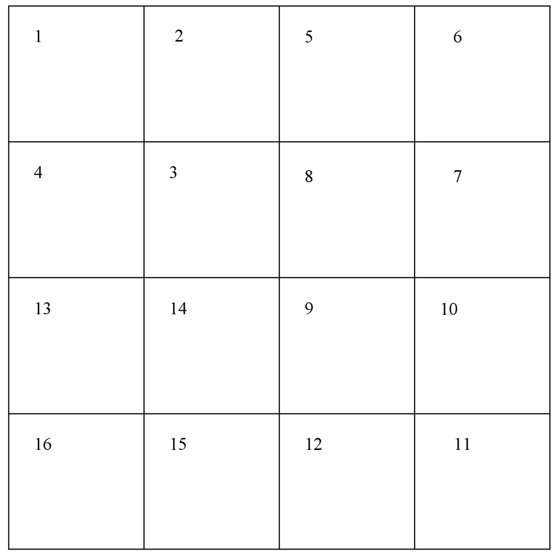

Fig. 9. Grids for the resolution $level_2$ in the example

$level_3$. Fig. 8 shows all grids and pixel points for $level_3$, the highest resolution level in this application. For the highest resolution level, the pixel point sequence number in a grid is the same as the grid number. The pixel points are numbered by calling algorithm 1 with parameters as $num\_grids(1, 4^3, 1, 2^3, 1, 2^3)$. Fig. 9 shows all grids of $level_2$ with grid sequence numbers. We use t to denote the period of time queries are sent to the sensor network and the value of a cycle for $level_2$ is $4t$. For $level_2$, the pixel points queried would be (1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61) at time t, (2,6,10,14,...,62) at time $2t$, (3,7,11,15,...,63) at time $3t$, (4,8,12,16,...,64) at time $4t$, and thus finish a cycle. We can see the pixel points for the first 4 times for $level_2$ combined are equal to the pixel points of the highest resolution level $level_3$.

An advantage of this scheme is the seamless connection of data retrieving between resolution levels. For example, if the $level_i$ speeds up its querying by 4 times, it becomes $level_{i+1}$. Users can use the period of querying as a parameter to control the resolution level of data retrieving.

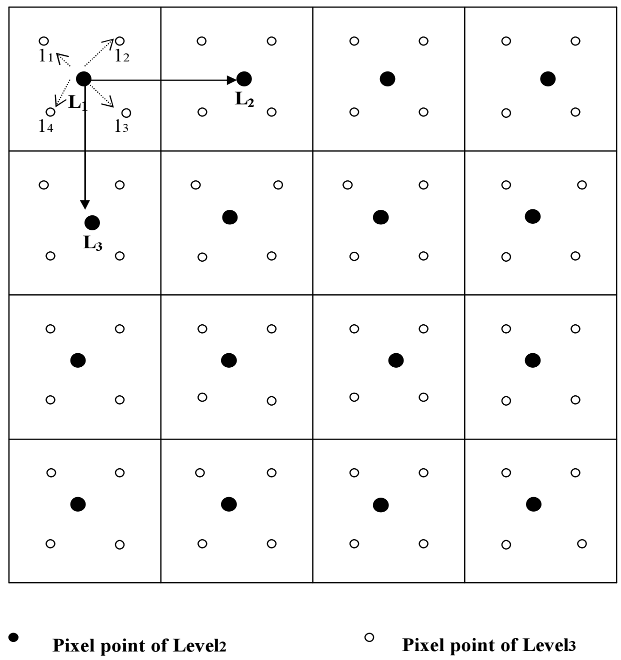

Fig. 10. An example of data-sensitive querying scheme

b. Data-sensitive Querying Scheme

It is always sufficient to represent the condition of the network field when querying in the highest resolution level. However, when data variation somewhere in the network is high, defining a fixed set of pixel points is not adequate to represent data with high fidelity for data retrieval in lower resolution levels. In an area with less data variation, a less number of pixel points is sufficient to represent the conditions with high fidelity than in an area with higher data variation. Therefore, we propose a data-sensitive querying scheme which retrieves more detailed data from places with higher data variations to keep higher fidelity for data retrieval in lower resolution levels.

In this scheme, after a query reaches a pixel point of a lower resolution level, beside being transferred directly to all children pixel points of the same resolution level, the query attached with an abstraction of the data of the current pixel point is forwarded to all pixel points of one higher resolution level within its grid. The data abstraction is to represent its own knowledge of the monitored phenomena, e.g., the temperature of the area around. A pixel point receiving such queries compares the attached data with its own data and report its data to the parent pixel point if the difference is above a threshold. Otherwise, the query is simply discarded. Suppose the pixel points are fixed at the center of the grids for a resolution level, Fig. 10 shows an example with two resolution levels, $level_2$ and $level_3$. When pixel point L_1 of $level_2$ receives queries, it not only forwards the query to L_2 and L_3, two of its children pixel points in $level_2$, but also forwards the queries attached with its own data to l_1, l_2, l_3 and l_4, all four pixel points of $level_3$ in its grid. Nevertheless, if there is no node inside a grid, the query reaching a home node outside the grid is not transferred further to pixel points of higher resolution levels in its grid.

In this manner, the data retrieved from regions with high data variation is more than that with lower data variation and keeps better fidelity for data retrieval in lower resolution levels.

4. Query and Data Forwarding

The connection among pixel points and the sink can be modeled as a connected, undirected graph G = (V,E), where V is the set including associated pixel points and the sink, E is the set of interconnections among pixel points and the sink. For each edge(u,v)$\in$ E, we have a weight w(u,v) specifying the cost (the distance between them) to connect u and v. We then wish to find a subset T $\subseteq$ E that connects all of the vertices and whose total weight

$$w(T) = \sum_{(u,v)\in T} w(u,v)$$

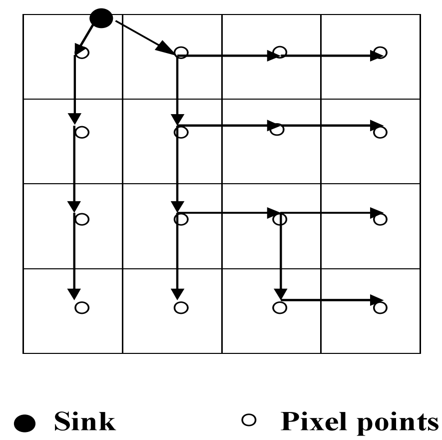

Fig. 11. The minimum-spanning-tree composed of pixel points and the sink

is minimized. We use Prim's minimum-spanning-tree algorithms to solve this problem [11]. Fig. 11 shows a minimum spanning tree found based on the algorithm.

For one resolution level, queries are always originated by the sink and dispatched to all pixel points along paths on the tree. A home node receiving a query makes copies and sends a query to each of its children pixel points. Data matched to the interest in the query is forwarded towards the sink along the same path of the query in reverse direction. At each inner node of tree, in-network aggregation may be conducted. For robustness to failures, an inner node may recruit all neighbors as replica.

C. Analysis

We analyze the energy efficiency of our scheme in comparison with flooding-based data gathering methods represented by Directed Diffusion. We use SMRDD to denote our method for the convenience of expression. Because communication is the most energy expensive operation for wireless devices, we use the communication overhead as the main criteria to compare the performance of our schemes with currently existing techniques.

For the convenience of analysis, we simplify the system model in which in-network aggregation is not considered since it can be applied to both methods. We propose a sensor network with N nodes uniformly deployed in a square field of area A. The communication overhead to flood an area is proportional to the number of sensor nodes in it, and to send a packet along a path by greedy geographical forwarding is proportional to the number of sensor nodes on the path. We use a to denote the side length of a gird for the highest resolution level in SMRDD, l_q to denote the size of a data packet which is either a query or an event. The distance between nodes is therefore $\sqrt{A}/(\sqrt{N}-1) \approx \sqrt{A}/\sqrt{N}$, *where* N *is a large number.*

The communication overhead is composed of two main parts: query dispatch from a sink to nodes and the transmission of data from nodes to the sink. We compare the performance in these two aspects between SMRDD in the highest resolution level and DD because they provide data in the same detail level.

First, we study the overhead for sending queries once across the entire network. For SMRDD, the total path length for query dispatch is the sum of all branch lengths in the minimum-spanning-tree which is at most $A/a^2 \times a = A/a$, suppose the sink is inside the network. The communication overhead is

$$E'_{smrdd} = \frac{\frac{A}{a}}{\frac{\sqrt{A}}{\sqrt{N}}} \times l_q = \frac{\sqrt{A}\sqrt{N}}{a} \times l_q \ . \tag{3.3}$$

Because DD uses flooding to send queries across the network, each time the communication overhead is

$$E'_{flooding} = N \times l_q \ . \tag{3.4}$$

Therefore,

$$\frac{E'_{smrdd}}{E'_{flooding}} = \frac{\frac{\sqrt{A}\sqrt{N}}{a} \times l_q}{N \times l_q} = \frac{\sqrt{A}}{a\sqrt{N}} = \frac{distance\ between\ nodes}{a} \ , \tag{3.5}$$

which means

$$\frac{E'_{smrdd}}{E'_{flooding}} < 1 \ \ (when \ \frac{distance\ between\ nodes}{a} < 1) \ . \tag{3.6}$$

From equation 3.6, it is obvious that, when the distance between nodes is smaller than the length of side of the grid (determined by the sensing range of sensors), SMRDD spends less energy than DD for query dispatch once. Actually SMRDD outperforms DD in query dispatch significantly because DD needs periodically broadcast queries to maintain the paths for data retrieval.

Next we compare the second component of communication overhead, the transmission of data to the sink. We use A' to denote the total area of the regions in which nodes have matched data. The number of nodes which are queried and, in response, send data to the sink in SMRDD is $n_s = A'/a^2$. The number of nodes which receive queries and correspondingly send data to the sink in DD is $n_d = A'/(A/N)$. $c\sqrt{N}$ is the average number of sensor nodes along the straight-line path from a data source to the sink ($0 < c \leq \sqrt{2}$). Because a data packet in SMRDD traverses a grid instead of straight-line path, the worst-case path length is increased by a factor of $\sqrt{2}$.

Suppose the average number of data packets retrieved from a node, which is denoted with p, is the same for both methods, the communication overhead for data transmission in SMRDD is

$$E''_{smrdd} = \frac{A'}{a^2} \times p \times \sqrt{2}(c\sqrt{N}) \times l_q \ , \tag{3.7}$$

while the communication overhead for data transmission in DD is

$$E''_{flooding} = \frac{A'}{\frac{A}{N}} \times p \times (c\sqrt{N}) \times l_q \ . \tag{3.8}$$

Therefore,

$$\frac{E''_{smrdd}}{E''_{flooding}} = \sqrt{2}(\frac{\sqrt{A}}{a\sqrt{N}})^2 = \sqrt{2}(\frac{distance\ between\ nodes}{a})^2 \ , \tag{3.9}$$

which means

$$\frac{E''_{smrdd}}{E''_{flooding}} < 1 \quad (when\ \frac{distance\ between\ nodes}{a} < 0.84)\ . \tag{3.10}$$

From equation 3.10, SMRDD is more energy efficient for data transmission than DD if the grid size is a little bit larger than the average distance between nodes even only one copy of the data is sent to the sink in DD. Actually in DD, multiple copies of data are flowing towards the sink upon being queried although one of them could be reinforced.

In sum, SMRDD is more efficient than flooding-based data gathering method when the density of nodes is relatively large in comparison to the sensing range of sensors for an application.

D. Simulation

We study the performance of SMRDD in the highest resolution level compared to DD via simulation in ns-2 [34] because they both retrieve data in the same resolution level. We use the CMU wireless extensions which includes full simulation of the IEEE 802.11 physical and MAC layer. Our simulations are for networks of nodes with 802.11 WaveLAN radios. The radio range is changed to 40 meters to make it closer to the real situation.

Three metrics are used for evaluation of performance. **Average energy consumption** is defined as the ratio of the total dissipated energy per node in the network to the number of packets successfully received by the sink. This metric defines the energy efficiency of the protocol. **Packet delivery success rate** is defined as the

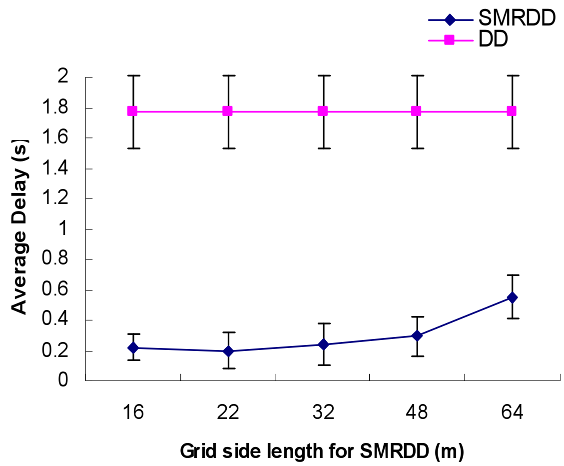

Fig. 12. Average delay of simulation

ratio of the number of data packets successfully received by the sink to the number of data packets sent by the data sources. This metric defines the effectiveness of data delivery. **Average delay** is defined as the average time between the moment a data packet is sent by a data source and the moment the sink receives the data packet. This metric defines the freshness of data packets.

We select a simple one sink simulation setting, in which sensor nodes randomly distributed in a 256*256m^2field. The sink is randomly located at lower left corner of the rectangle sensor field while the data sources are all nodes located at the upper right corner region with $\frac{1}{4}$ of the area of the sensor field which send packets periodically to the sink upon being queried. Queries are sent across the network because the data sources are not known in advance and could be anywhere randomly across the network.

We notice that the average energy consumption and average delay increases a little when the grid size increases in SMRDD, as shown in Fig. 12 and Fig. 13. It is caused by underlying routing protocol OD-GPSR. OD-GPSR is a data driven, on demand stateless protocol. The first several packets in each flow experience higher

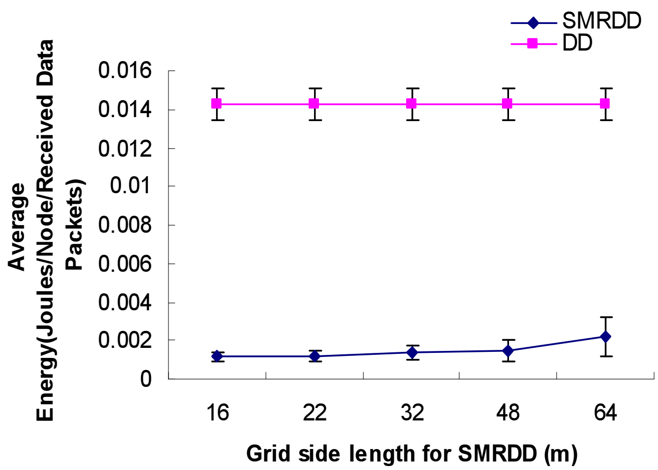

Fig. 13. Average energy consumption of simulation

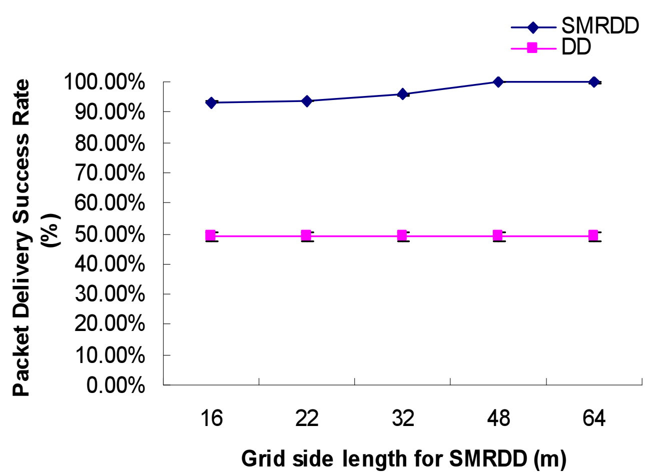

Fig. 14. Packets delivery success rate of simulation

delay because the packets have to wait for neighbor information for making routing decision if the current nodes have no neighbor information. For larger grid size, the number of data packets is less because less home nodes are queried. As a result, the average delay is higher than smaller gird sizes. It is the same situation for average energy consumption. Although the total energy consumption which includes querying overhead and data transmission overhead absolutely decreases for larger grid sizes, the average energy consumption is a little bit larger due to less data packets generated.

From Fig. 12, Fig. 13, and Fig. 14, SMRDD performs much better than DD in all three metrics. In the simulation, we notice that the packet number for SMRDD is significantly less than DD because of several reasons. First, under SMRDD, on-demand geographical routing support is used to eliminate flooding. The control packets are further reduced in OD-GPSR by prohibiting beacon exchange among neighbors in areas without data flows. Second, as the grid size increases, less nodes are queried and therefore less data from data sources is transmitted. Consequently, the success delivery rate increases because of less possible collisions incurred by less traffic across the network, as shown in Fig. 14.

E. Contributions

For applications in which users routinely retrieve data from the sensor networks such as temperature monitoring, we propose a scheme to provide spatial-based multiple resolution data dissemination services to meet various detail requirements. In contrast with data dissemination methods directly querying on nodes, our scheme sends queries to locations, called pixel points, across the sensor network. The number and density of pixel points are application-based, depending on sensors' sensing range for tasks. In addition to eliminating query flooding and high overhead for data-centric storage, this method elicits less redundant data and achieves further energy efficiency.

In comparison with flooding-based data dissemination techniques, this scheme has following advantages in terms of energy efficiency:

- Query dispatch involves necessary nodes instead of using broadcasting to all nodes. Nodes not involved in the process of data retrieval keep silence to save valuable energy.
- Decrease the transmission of redundant data without loss of fidelity by only retrieving data from pixel points in the highest resolution level.
- Provide flexible multiple resolution data dissemination with which users are able to retrieve data from different subareas with various resolution requirements simultaneously and efficiently.

Analysis shows that this scheme is best suitable for large-scale wireless sensor networks with relatively high density of nodes in terms of energy efficiency.

CHAPTER IV

A REACTIVE MULTI-RESOLUTION DATA DISSEMINATION SCHEME

A. Introduction

Fig. 15 shows an example of the other category of sensor network applications we focus on. In this type of applications, sensor networks are deployed to detect some mobile objects such as birds or animals. Similarly, in the real world, phenomena monitored by sensor networks in environments often show up unpredictably. Correspondingly data sources(a *data source* is composed of nodes around a stimulus with detected data) are generated in an unexpected manner across the networks.

This type of application has some similar properties as the previous one. For instance, it is always the sink which initiates the data retrieval and data is to be sent to the sink from data sources upon receiving queries; data redundancy is inevitable for a large-scale wireless sensor network with high density of nodes and decreasing transmission of redundant data would significantly improve the performance in energy efficiency; less detailed data is acceptable for such applications under some circumstance for the purpose of energy efficiency.

Nevertheless, this type of applications has its own unique properties. The most important one is that the locations of data sources and the time they appear are unknown in advance and unpredictable. A data source could emerge at any time and the location of the data source could be anywhere across the network. Because the emergence of data sources are unpredictable, how to locate the data sources and send queries to them efficiently is a challenge for efficient data dissemination. Flooding is an option to solve this problem, which, however, is not energy efficient. Data-centric storage technique is an option to handle this problem without flooding, but

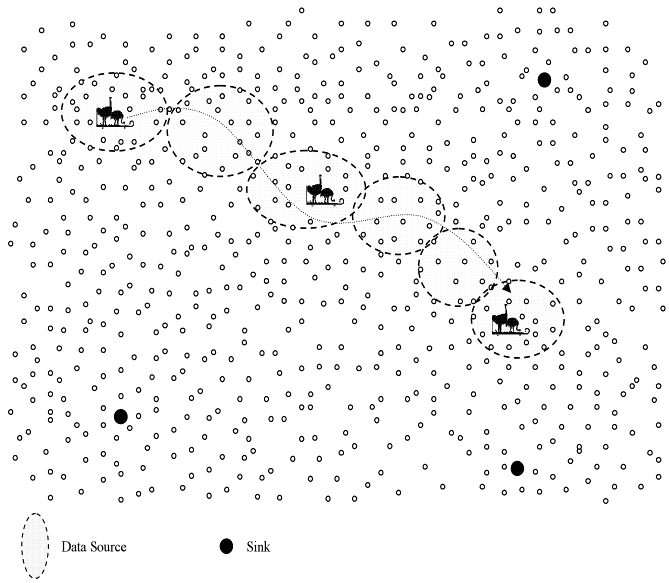

Fig. 15. A sensor network application detecting animals

it is not efficient when a large volume of data is to be stored elsewhere. Therefore, how to minimize the data to be stored has much effect on the performance of data dissemination when taking data-centric storage techniques.

Multi-resolution data has two advantages in this scheme from the perspective of energy efficiency. First, it decreases the transmission of redundant data. Second, it accomplishes energy efficiency by transmission of less data at the cost of data fidelity for data dissemination in lower data resolution levels. Therefore, an efficient and flexible scheme to provide multi-resolution data dissemination in correspondence to the resolution requirement is very important for efficient data dissemination. As an important component of the scheme, how to model the multi-resolution data and correspondingly how to design an algorithm to efficiently pick a subset of data from data sources in correspondence with the data resolution requirement is essential for multi-resolution data dissemination.

In this chapter, we propose an efficient data dissemination scheme for such applications to accomplish the goals described above. This scheme takes advantage of both local-storage and in-network data centric storage techniques for energy efficient data dissemination. Since flooding is the very expensive in terms of energy consumption, the scheme aims at limiting the flooding as much as possible. Furthermore, data-centric storage techniques are taken to eliminate query flooding while the scheme tries to minimize the amount of data for storage elsewhere for energy efficiency.

Our scheme accomplishes further energy efficiency by reducing dissemination of redundant data. Moreover, this scheme provides multi-resolution data dissemination for energy efficiency. In this scheme, generated data is stored locally while data sources register to *registration points* (geographical locations dispersed across the network) using data centric storage techniques. During the formation upon detection of stimuli, a data source initiates a localized algorithm to select a node as the leader to register to the nearest registration point on behalf of the data source. When sinks solicit data from the network, queries with resolution specifications are sent to and stored at all registration points which forward them to all matched data sources. When a data source receives a query, a set of nodes in correspondence with the resolution specification in the query is selected to report data to sinks. The nodes with data of high redundancy are prevented from being queried and thus the dissemination of redundant data is reduced. For load balance, tasks in the sensor network could have different registration points dispersed across the sensor network.

To support the delivery of packets to a particular location, we use On-demand GPSR(OD-GPSR), a modified version of Greedy Perimeter Stateless Routing(GPSR), which is customized for sensor networks. The design of OD-GPSR is stated in detail in Chapter V.

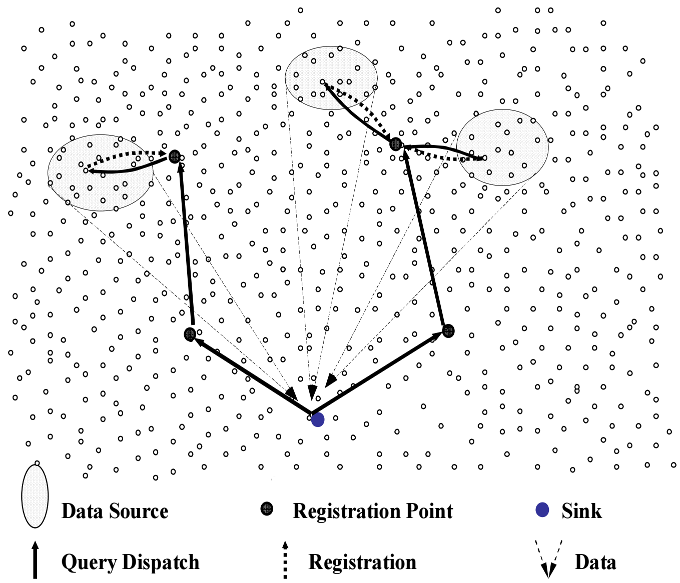

Fig. 16. The architecture of the reactive multi-resolution data dissemination scheme

B. Methodology

We assume a large scale sensor network composed of static energy-constrained sensor nodes with uniform transmission ranges and the density of nodes in the network is relatively high. Each node has the knowledge of its own location.

In this scheme, we use the same application-oriented spatial-based multi-resolution data model as that in the scheme described in Chapter III. However, the implementation is different. We will present in detail later the algorithm to select a set of node in correspondence with the resolution requirements based on the locations information of nodes inside a data source and the sensing range of nodes.

1. Overview of the Scheme

As shown in Fig. 16, this scheme is composed of four main components: a data-source-forming algorithm to elect a leader within a data source which represents the data source for registration, a hierarchical multi-resolution data querying scheme, data delivery from nodes to sinks, and an underlying geographic routing protocol which routes a packet to the node geographically closest to the target location.

Composed of nodes around the stimuli holding detected data, a data source is formed upon detection of a stimulus, during which a leader is elected to represent the data source. The leader registers on behalf of the data source to the nearest registration point with information such as the location of the leader, the type of the data, the birth date of the data source as well as an abstraction of the data such as the maximum and minimum value of data which could be used for narrowing the range of querying or multi-dimensional data querying.

Query dispatch consists of three phases: First, queries originated from sinks are sent to all registration points calculated by using geographic hashing functions [39]. In the second phase, a registration point makes copies of the query and forwards them to all matched data sources since it has records of all data sources in its own territory. Inside a query a parameter of resolution level is included. Each resolution level corresponds to a distance value which is used to pick a subset of nodes from the data source using the node-selection algorithm. The mapping between the value and resolution level is up to applications except the highest resolution level, in which the distance value is determined by the sensing range of the sensor device used for the task. The third phase starts when the leader of a data source receives a query and then use a node-selection algorithm to pick a set of nodes inside the data source to be queried.

The location of a sink is included in the query it dispatches. The nodes being queried either send data directly to the sink or send data to its leader for aggregation before being sent to the sink.

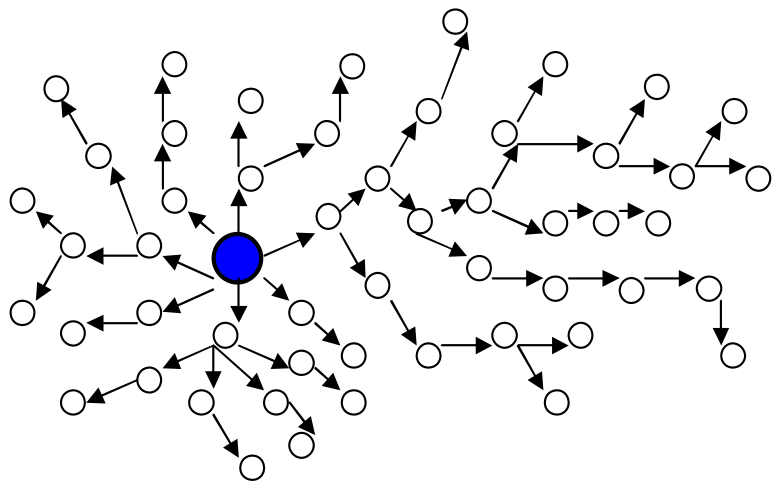

Fig. 17. Data source forming process

2. Data Source Forming

Composed of nodes around the stimuli holding detected data, a data source is formed upon detection of a stimulus. In the process, one leader is elected to represent the data source. This localized algorithm is shown in Fig. 17. When a node detects stimuli and has no corresponding leader records, it sets a timer with a random value between 0 and a value T. When this timer expires, it broadcasts a message declaring itself as the leader. This message includes following information: its location and the type of phenomena. The location of the leader instead of identifier is used to represent the leader and its data source. Other nodes in the data source keep a record of the leader upon receiving the message. A node in the data source will discard duplicated copies of the message if it already has a leader record of the same type of phenomenon. After the election process, the leader elected registers on behalf of the data source to the nearest registration point with information such as the location of the leader, the type of the data, the birth date of the data source as well as an abstraction of the data such as the maximum and minimum value of data which could be used for narrowing the range of querying or multi-dimensional data querying.

Following is the pseudo-code of the algorithm to elect a leader for a data source.

```
Upon detecting phenomena for a node
if(already has a leader record for this phenomenon)
    return;
else
    set an election timer with a random value of time in {0, T}

when the election timer expires,
    if(already has the leader record for the phenomenon)
        return;
    else {
        send a message declaring itself as the leader to all neighbors;
        set the timer for registration;
    }
when the timer for registration expires,
    register to the nearest registration point;

Upon receiving a declaration message for a node
if(not has this type of data) {
     inform the source node that it is the boundary;
     discard the message;
     return; /*reach the boundary of the data source*/
} if(has a leader record for this type of data && the leader is different) {
     inform the source node the boundary together with its leader record;
     discard the message;
     return;
}
if(has a leader record for this type of data) {
     save the children information if I am the parent;
     discard the message;
     return;
}
if(a timer is set for leader declaration)
     remove the timer;
save the leader message;
save the parent information;
broadcast the message to all neighbors;
```

Algorithm 3. The algorithm for Data source forming

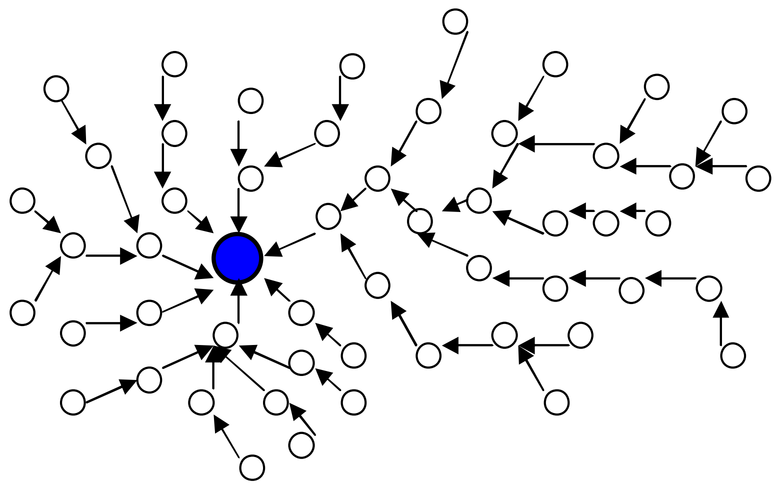

Fig. 18. The leader collects information from members

3. Collecting Information from Members

In the next phase, members in a data source reports information, such as its location and an abstraction of its data, to the leader, as shown in Fig. 18. During the process of data source forming, each member remembers the neighbor node which sends the leader message first. This neighbor is the next hop on the path towards the leader. The node will broadcast its parent along with the leader message. Therefore, the parent is able to know all of its children. In this manner a tree composed of all members of a data source is formed with the leader as the root and each member has a unique path towards the leader.

The process for all members in a data source to report data is initiated by the nodes on the boundary. When a node receives a leader message and finds itself without that type of data, it reminds the message source that it reaches the boundary. The boundary nodes which have no children will send its own information towards its parent in the tree after a while. Its parent will send its own data together with this data to its parents after receiving information from all children. The process continues

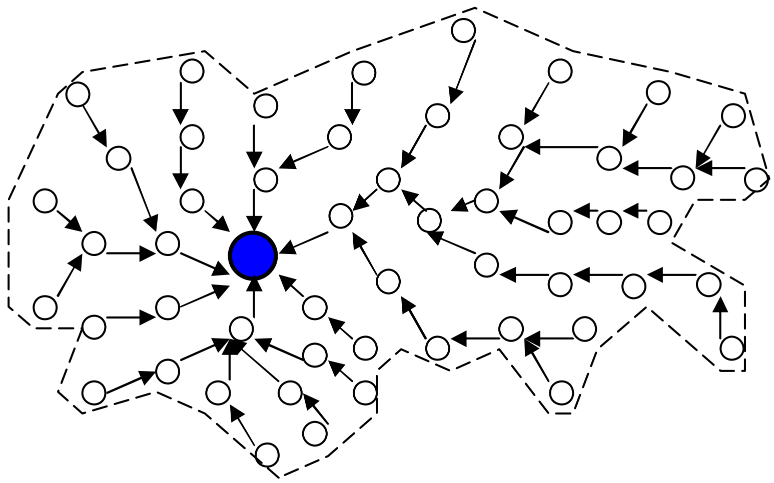

Fig. 19. The first round for members to report information to their leader

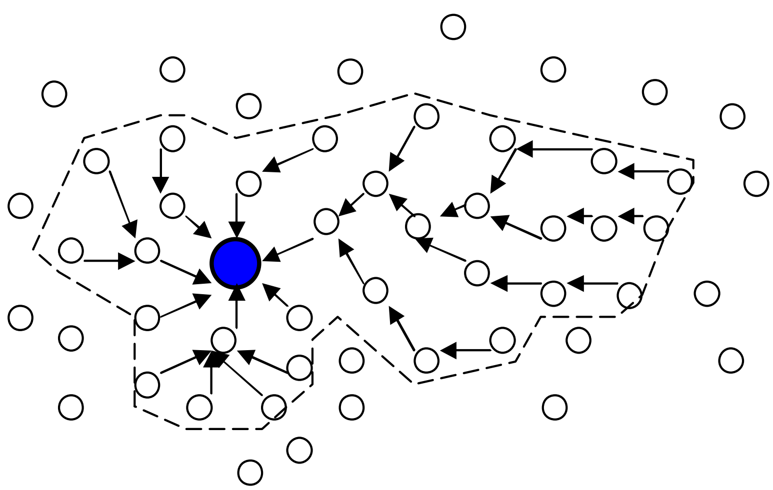

Fig. 20. The second round for members to report information to their leader

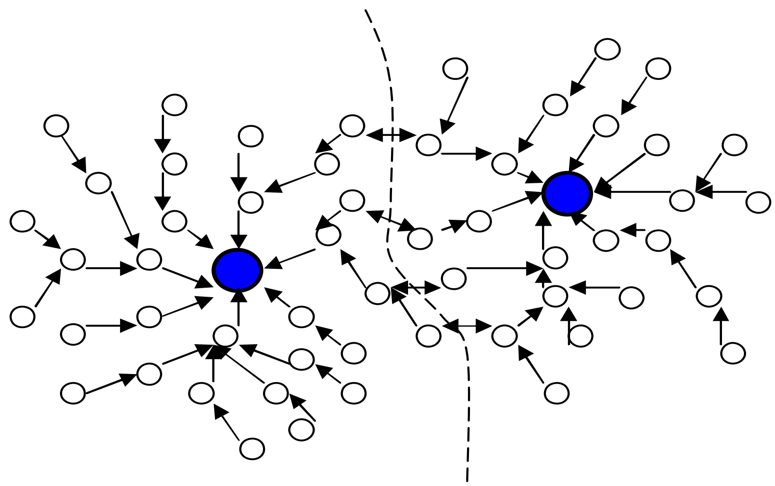

Fig. 21. The leaders of neighbor data sources know each other

until the data reaches the root, the leader. The first round of this process is shown in Fig. 19 and the second round is shown in Fig. 20. The process is synchronized by data being passed from leaf nodes towards the leader.

If more than one data sources are connected, the process enables leaders of neighbor data sources to know each other. This process is shown in Fig. 21. Some boundary nodes of the neighbor data sources are neighbors with each other and are able to exchange with each other the leader information. This information is included in the data sent to the leader and, as a result, leaders of neighbor data sources are known each other. From these leaders, one is elected to register on behalf of all data sources. For example, the leader with the largest y coordinate is selected.

Note that this phase is optional in the design of the sensor networks in that, if the sensor network is designed not to provide multiple resolution data, this phase is not necessary.

Query dispatch consists of three phases: First, queries originated from sinks are sent to all registration points which can be got by using geographic hashing functions [39]. The number of registration points depends on the application and tasks could have different registration points dispersed across the network. This design has twofold consideration. First, selecting different set of registration points for tasks achieves load balance. Second, using more than one registration points for a task lessens the *hot spot* problem in which some nodes dies much more quickly than other nodes in the network for out of the energy caused by the overwhelming access. The dispatch of queries from a sink to registration points goes along paths of a tree which is rooted at the sink and composed of all registration points with the least total length of all edges.

The second phase is from registration points to all matched data sources. A registration point has records of all data sources in its own territory and is able to make copies of the query and send them to all matched data sources. Inside a query a parameter of resolution level is included. Each resolution level corresponds to a value which is used to select a subset of nodes from the data source using the *node-selection* algorithm. The mapping between the value and resolution level is up to applications except the highest resolution level, in which the value is determined by the sensing capability of the sensor device used for a task.

The third phase starts when the leader of a data source receives a query. The leader selects a set of nodes inside the data source to be queried based on the location information of all nodes and the resolution requirement. Queries are sent to those selected nodes along paths in the tree from which they were collected but in reverse direction.

Following is a brief description of the algorithm to select a set of nodes for a resolution level based on the spatial information of nodes in a data source. The algorithm is run by the leader since all location information of nodes are stored here. First, the following function is called to check if the sensing area of the leader is totally covered by other adjacent nodes which are not selected. If it is, this node is not selected. Otherwise, the node is to be queried. Then randomly pick another node to check until all nodes are checked. Following is the pseudo-code of the function.

```
void node_selection() {
   set the status of all nodes as undetermined;
   sort all nodes based on the coordinates;
   radius = mapping_of_resolution();
   i = 0;
   while(i < the number of nodes in the data source) {
         node = nodes[i];
         if( covered_by_neighbors(node, radius) ) {
              set status of the node as not-selected;
         }
         else {
              set status of the node as selected;
         }
   }
}

/* This function is to determine whether the sensing range of the current node*/
/* can be covered by those neighbors with status of null and being queried */
boolean covered_by_neighbors(node, radius) {
   exclude all neighbors with status of not-selected;
   /* the circle with my location as center and the radius*/
   uniformly sample my sensing range with some density;
   for each point inside my sensing range {
       if(the distance to all neighbor locations > radius)
           return false;
   }
   return true;
}
```

Algorithm 4. Node-selection algorithm in a data source

We use an example to show how this algorithm works. Fig. 22 shows the result after running the algorithm on a uniformly deployed sensor network when the sensing

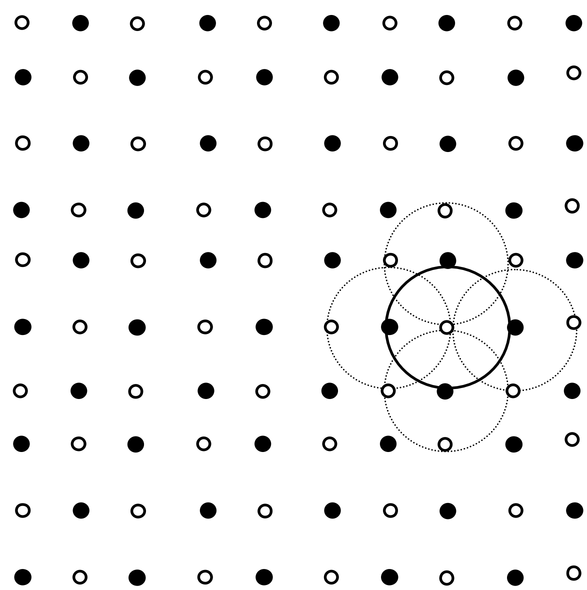

Fig. 22. Node selection result when r >= d

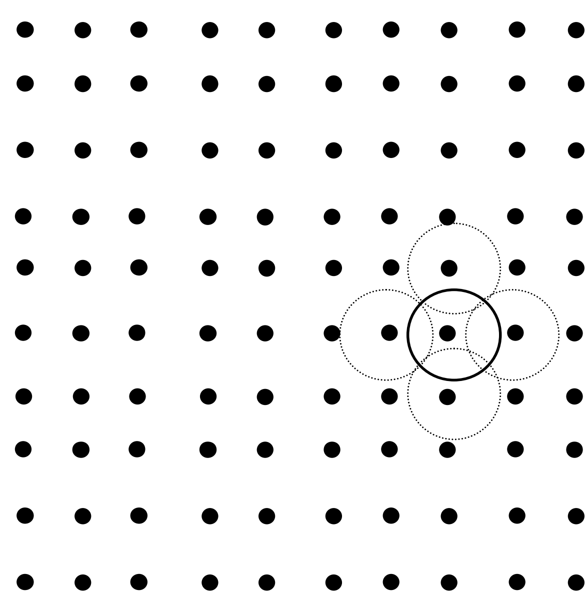

Fig. 23. Node selection result when r < d

range is large(radius $r >=$ distance between nodes d). The filled black dots, the number of which is approximately half of the nodes, are selected nodes while others are not selected because their sensing ranges are completely covered by those selected nodes. In Fig. 23, because the sensing range of any nodes can not be completely covered by neighbor nodes, every node is selected.

5. Data Delivery

There are two options for data delivery. Since the location of a sink is included in the query it dispatches, the nodes being queried are able to send data directly to the sink via our geographical routing protocol, OD-GPSR, which is a modified version of GPSR which routes packets to a particular geographical location.

The other option for data delivery is that nodes being queried send the data to the leader first, where data is aggregated. Then the data is sent to the sink by the leader. The advantage of this method is that the number of packet could be significantly decreased after aggregation and thus communication cost is reduced. The disadvantage is that it incurs hot-spot problem in that nodes around the leader could die much faster than other nodes due to massive communication cost.

C. Analysis

We analyze the energy efficiency of this scheme in comparison with flooding-based data gathering methods. We still use SMRDD to denote this method for the convenience of expression. Because communication is the most energy expensive operation for wireless devices, we use the communication cost as the main criteria to compare the performance of our schemes with current techniques.

We assume an application in which there is only one registration point. Just as the analysis in Chapter III, for the convenience of analysis, we simplify the system model in which in-network aggregation is not considered. We propose a sensor network with N nodes uniformly deployed in a square field of area A. The communication overhead to flood an area is proportional to the number of sensors in it, and to send a packet along a path by greedy geographical forwarding is proportional to the number of sensor nodes on the path. We use l_q to denote the size of a data packet which is either a query or an event. The distance between nodes is therefore $\sqrt{A}/(\sqrt{N}-1) \approx \sqrt{A}/\sqrt{N}$, *where* N *is a large number.*

1. Case 1: No Data Existing for Retrieval

Since it is always the sink which initiates the data retrieval from the networks, there is possibility that no interested data exists when the queries are sent. We compare the overhead for such cases first. $c\sqrt{N}$ is the average number of sensor nodes along the straight-line path from the sink to the registration point and from a registration point to a leader($0 < c \leq \sqrt{2}$). The worst-case path length is increased by a factor of $\sqrt{2}$.

Because flooding-based techniques use flooding to send queries across the network, each time the overhead is

$$E_{flooding} = N \times l_q \ . \tag{4.1}$$

For SMRDD, the overhead is

$$E_{smrdd} = c\sqrt{N} \times l_q \ . \tag{4.2}$$

Therefore,

$$\frac{E_{smrdd}}{E_{flooding}} = \frac{c\sqrt{N} \times l_q}{N \times l_q} = \frac{c}{\sqrt{N}} < \frac{\sqrt{2}}{\sqrt{N}} \;, \tag{4.3}$$

which means

$$\frac{E_{smrdd}}{E_{flooding}} < 1 \quad (when\ N > 2) \;. \tag{4.4}$$

From equation 4.3, it is obvious that this scheme saves much energy for cases when the sink initiates data retrieval while there is no interested data existing.

2. Case 2: Data Existing for Retrieval

For SMRDD, the communication overhead is composed of two main parts: the overhead for querying which includes data source forming and registration, query dispatch from a sink to nodes, and the overhead on the transmission of data from nodes to the sink. We compare the performance in these aspects between SMRDD and flooding-based techniques. We consider two options of SMRDD, data retrieval in the highest resolution level and using all-node-querying option(without resolution requirement), both of which provide data in the same detail level as flooding-based techniques.

First we consider the all-node-querying option for SMRDD. Since the transmission of data from nodes to the sink is the same for this option of SMRDD and flooding-based techniques, we compare the the overhead for querying only.

Suppose there are m data sources, the communication overhead for data source registration is $m \times c\sqrt{N}$. We use A' to denote the total area of all data sources and then the number of nodes in data sources is $N \times A'/A$. This is the communication overhead for data source forming because local flooding is used for leader election and data source forming. The communication overhead for sending queries to every node in all data sources is $(m+1) \times c\sqrt{N} + N \times A'/A$.

Therefore,

$$E'_{smrdd} = ((2m+1)c\sqrt{N} + 2N\frac{A'}{A}) \times l_q \ . \tag{4.5}$$

Because flooding-based techniques use flooding to send queries across the network, each time the communication overhead is

$$E'_{flooding} = N \times l_q \ . \tag{4.6}$$

Therefore,

$$\frac{E'_{smrdd}}{E'_{flooding}} = (2m+1)\frac{c}{\sqrt{N}} + 2\frac{A'}{A} < (2m+1)\frac{\sqrt{2}}{\sqrt{N}} + 2\frac{A'}{A} \ . \tag{4.7}$$

From equation 4.7, this scheme is more efficient than flooding-based techniques when being applied to large-scale sensor networks and the data source area is a small proportion of the network area(less than half of the area of the sensor field). Also, it is suitable for cases when the number of data sources is small in comparison to the number of sensor nodes.

Next, we consider the data retrieval with resolution requirement and the resolution is in the highest resolution level. Similarly, the communication overhead for data source registration is $m \times c\sqrt{N}$. The communication overhead for sending queries is $(m+1) \times c\sqrt{N} + N \times A'/A$. The communication cost for data source forming is $2N \times A'/A$.

Therefore, the communication overhead for querying in total is

$$E'_{smrdd} = ((2m+1)c\sqrt{N} + 3N\frac{A'}{A}) \times l_q \ . \tag{4.8}$$

As a result,

$$\frac{E'_{smrdd}}{E'_{flooding}} = (2m+1)\frac{c}{\sqrt{N}} + 3\frac{A'}{A} < (2m+1)\frac{\sqrt{2}}{\sqrt{N}} + 3\frac{A'}{A} \ . \tag{4.9}$$

From equation 4.9, if we want this scheme to be more efficient than flooding method in querying process, the area of data sources should be less than third of the entire area of sensor field and the number of data sources should be small relative to the number of sensor nodes.

Another important component of communication overhead is the transmission of data to the sink. We use r to denote the value of the radius for the highest resolution level in SMRDD, which is used to select a subset of nodes in a data source to be queried. We use d to denote the distance between nodes. $c\sqrt{N}$ is the average number of sensor nodes along the straight-line path from a data source to the sink $(0 < c \leq \sqrt{2})$. Suppose the average number of data packets retrieved from a node, which is denoted with p, is the same for both methods, the communication overhead for data transmission in SMRDD is

$$E''_{smrdd} = N\frac{A'}{A} \times p \times (c\sqrt{N}) \times l_q \quad (if\ r < d)\ , \tag{4.10}$$

or

$$E''_{smrdd} <= \frac{N}{2}\frac{A'}{A} \times p \times (c\sqrt{N}) \times l_q \quad (if\ r >= d)\ , \tag{4.11}$$

while the communication overhead for data transmission in flooding method is

$$E''_{flooding} = N\frac{A'}{A} \times p \times (c\sqrt{N}) \times l_q\ . \tag{4.12}$$

Therefore,

$$\frac{E''_{smrdd}}{E''_{flooding}} = 1 \quad (if\ r < d) \tag{4.13}$$

or

$$\frac{E''_{smrdd}}{E''_{flooding}} <= \frac{1}{2} \quad (if\ r >= d) \tag{4.14}$$

From equation 4.14, this option is much more efficient than flooding method in data transmission when the node density is high relative to the sensing range.

Finally we compare the overall communication overhead when $r >= d$.

Because

$$\begin{aligned} \frac{E_{smrdd}}{E_{flooding}} &= \frac{E'_{smrdd} + E''_{smrdd}}{E'_{flooding} + E''_{flooding}} \\ &<= \frac{(2m+1)c\sqrt{N} + 3N\frac{A'}{A} + \frac{N}{2}\frac{A'}{A} \times p \times c\sqrt{N}}{N + N\frac{A'}{A} \times p \times (c\sqrt{N})} \ (if\ r >= d) \\ &= \frac{1}{2} + \frac{(2m+1)c\sqrt{N} + 3N\frac{A'}{A} - \frac{N}{2}}{N + N \times \frac{A'}{A} \times p \times c\sqrt{N}} \ , \end{aligned} \tag{4.15}$$

if we want

$$\frac{E_{smrdd}}{E_{flooding}} < 1 \qquad (\ r >= d)\ , \tag{4.16}$$

$$\frac{c(2m+1)}{\sqrt{N}} + (3 - \frac{pc\sqrt{N}}{2})\frac{A'}{A} < 1 \quad (\ r >= d)\ . \tag{4.17}$$

From equation 4.17, it is obvious that our scheme is more efficient than flooding-based techniques when being applied to large-scale sensor networks with high node density for cases where the number of data sources is small relative to the number of sensor nodes. This option(with resolution requirement) is best suitable for cases where the average number of data packets to be retrieved is large.

D. Simulation

We simulate our scheme and Directed Diffusion in ns-2 to study the performance. We use the CMU wireless extensions which includes full simulation of the IEEE 802.11 physical and MAC layer. Our simulations are for networks of nodes with 802.11

WaveLAN radios. The radio range is changed to 40 meters to make it closer to the real situation. We select a simple simulation setting, in which there are 2 sinks and sensor nodes randomly distributed in a 256*256m^2field. We use two registration points and two data sources. Each data source is a set of nodes in a circle area with radius of 40m. Each simulation run lasts for 1000 seconds, and each result is averaged over several random network topologies. A queried node generates one data packet every 5 seconds after receiving queries. Each data packet has 64 bytes. For OD-GPSR, the beacon interval is set as 5 seconds.

Three metrics are used for evaluation of performance. **Average energy consumption** is defined as the ratio of the total dissipated energy per node in the network to the number of packets successfully received by the sink. This metric defines the energy efficiency of the protocol. **Packet delivery success rate** is defined as the ratio of the number of data packets successfully received by the sink to the number of data packets sent by the data sources. This metric defines the effectiveness of data delivery. **Average delay** is defined as the average time between the moment a data packet is sent by a data source and the moment the sink receives the data packet. This metric defines the freshness of data packets.

From the simulation result shown in Figs. 24, 25, and 26, it is obvious that our scheme performs much better than DD in these three metric. The first reason, under our method, instead of all nodes are queried as DD does, only a subset of nodes, which have data with less redundancy, are queried. The more the sensing range is, the less number of nodes are queried and thus less packets are transmitted across the network. As a result, the performance is better. Furthermore, DD keeps refreshing data delivery paths by broadcasting queries periodically and maintains multiple paths from data source to the sink although it can reinforce some of them. These operations incur much traffic which in turn leads to high possibility of interference on the data

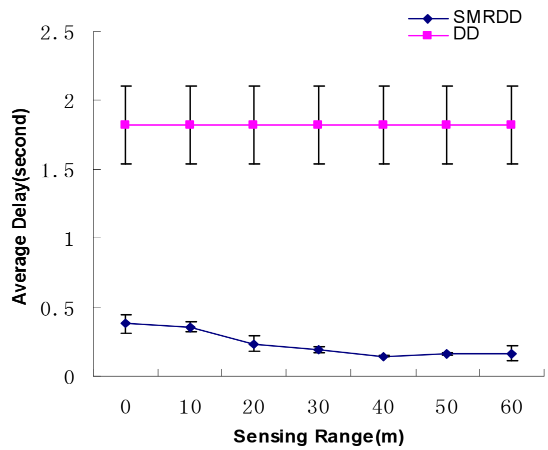

Fig. 24. Average delay

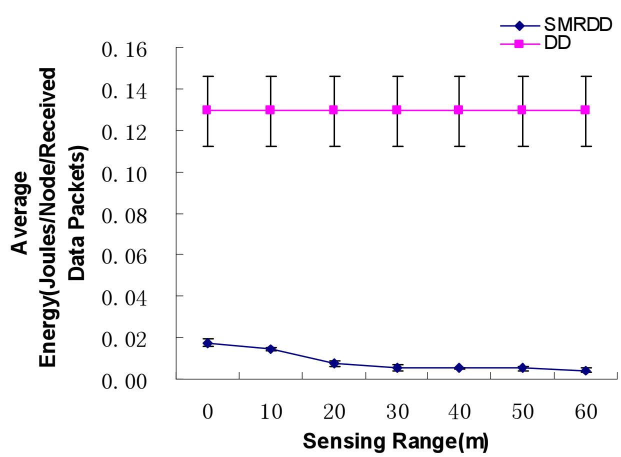

Fig. 25. Average energy consumption

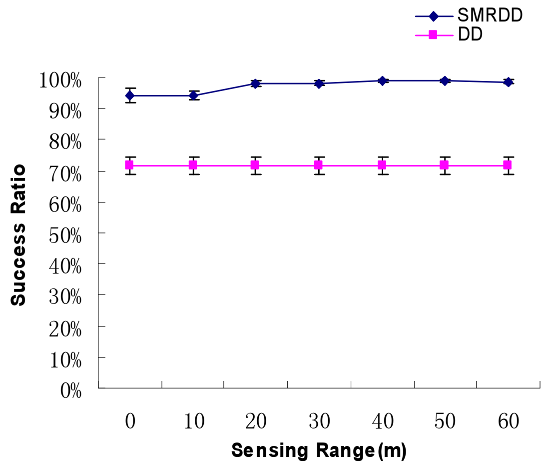

Fig. 26. Packets delivery success rate

forwarding and incurs higher packet delay and lower packet delivery rate.

E. Contributions

For applications such as detection of mobile objects in sensor networks, a reactive data dissemination scheme is proposed which achieves energy efficiency by eliminating querying flooding and reducing redundant data transmission. Furthermore, our scheme is capable of providing data retrieval of multiple levels of detail for some applications. The underlying geographical routing protocol is customized for sensor network for energy efficiency. In comparison with data-centric techniques, it significantly decreases the energy cost on data storage. Moreover, our scheme is capable of providing data retrieval of multiple levels of detail for some applications.

In comparison with flooding-based techniques, our scheme has following advantages in terms of energy efficiency:

- Query dispatch involves necessary nodes instead of using broadcasting to all nodes across the network;

- Decrease the transmission of redundant data without loss of fidelity by only retrieving data from pixel points in the highest resolution level;
- Provide data dissemination in multiple resolution levels to accomplish energy efficiency at the cost of data fidelity in lower resolution levels.

In comparison to the data-centric technique, our scheme has following advantages in terms of energy efficiency:

- Minimize the data to be saved elsewhere in storage points so that reduce the energy cost and lessen the hot spot problem significantly;
- Decrease the transmission of redundant data to the storage point so that save energy.

Analysis shows that this scheme is more efficient than flooding-based technique when it is applied to a large-scale sensor network with high node density relative to the sensing range of sensors for the task.

CHAPTER V

CUSTOMIZING GPSR FOR WIRELESS SENSOR NETWORKS

A. Introduction

Large-scale wireless sensor networks are composed of a large number of small and cheap nodes capable of wireless communication and significant computation. Traditional Internet network technologies cannot be directly applied to sensor networks. For example, a sensor node may not need an identity [12]. Spatial location plays an important role in sensor networks [22]. As in our design, both queries and events are sent towards locations instead of nodes directly. To support the delivery of packets to a particular location, a strongly geographic routing protocol is required in which the destinations of packets are marked with locations instead of node identifies like IP address. Packets are routed to the home node of the target location, the node geographically closest to the destined location.

As a strongly geographical routing protocol allowing nodes to send packets to a particular location, GPSR [26] is holding promise in providing routing support in wireless sensor networks. For instance, many recent research works on in-network data-centric storage such as [16], [19], [31], [39] build applications atop GPSR. However, because GPSR is not originally designed for sensor networks, several problems are required to be solved before it is applied in sensor networks [6], [7].

- In sensor networks, packet destinations are often marked with locations instead of identifiers like IP addresses and packets finally reach the node geographically closest to the destination, the *home node* of the target location. GPSR is designed to mark the destination with node identifier, IP address, which is not suitable for sensor networks.

- For sensor networks, GPSR is not efficient in terms of energy, one of the most valuable resources in sensor networks. GPSR is a proactive protocol under which each node is required to periodically send beacons to neighbors to update their status and location information, even when there is no traffic around. Although a beacon packet has very small size and is transmitted for one hop distance only, considering potentially the large number of nodes and long time of working duration, the energy consumption is considerable.

- *Data consistency* problem, which means the data retrieved from a location in sensor networks should be consistent with data sent to the same location, becomes a challenge due to the dynamic nature of sensor networks. Sensor nodes including the current home nodes for locations may often fall into disfunction state or even die due to hardware failure and energy exhaustion, or are intermittently reachable as a result of the impact of various factors, e.g., environmental effects.

Based on the implementation of GPSR, we propose *On-demand GPSR*(OD-GPSR), a data-driven geographical routing protocol. In OD-GPSR, the destination in a packet is identified with location instead of node IP address. Packets are routed to the node nearest to the target location(i.e. the home node for that location). In addition, OD-GPSR works more efficiently than GPSR under the same circumstance. Following is a briefly explanation on how OD-GPSR works.

Under OD-GPSR, only those nodes with data flowing over solicit location information from neighbors in support of routing decision. As a result, unnecessary communication between neighbors is eliminated and valuable energy is saved. When a node needs to forward packets but has no neighbor information, the node caches the packets first and then broadcast a one-hop *beacon-request* packet to all neighbors seeking neighbor information. In response, neighbors send back *beacon* packets including location information.

The routing decision for packets is made using the same algorithm as of GPSR but based on different neighbor information. Packets are forwarded greedily whenever possible based on neighbor information. The packet is always forwarded to the neighbor geographically closest to the target location in *greedy* mode. When a packet reaches a dead end with no closer neighbor, the packet switches to *perimeter forwarding* mode and uses right-hand rule to circumnavigate the void in a planarized network graph.

If a packet reaches a node whose distance to the destination is within half of its radio range and all neighbors are further to the target location than itself, this node is recognized as the home node. Otherwise, after a packet takes a tour of the enclosed perimeter around the target location and returns to the closest node to the destination, the node is recognized as the home node for that location. To maintain data consistency and improve robustness to node failures, OD-GPSR has the home node of a location recruit all neighbors as replica to cope with the dynamic property of sensor networks. The home node keeps broadcasting refresh packets to refresh the timers on neighbors. If the home node dies, a replica node will transfer data to the new home node after its timer expires. Also the current home node periodically sends special packets targeted to the represented location to check the existence of the possibly newly emerged closest node to the location.

We evaluate the performance of OD-GPSR through simulations in comparison with the GPSR version used in [39]. Results show that OD-GPSR has better performance in energy efficiency and packet delivery success rate at the cost of a little more packet delivery delay.

B. Methodology

We consider a network of *static* (e.g. immobile) energy-constrained sensors that are deployed over a flat region with each node knowing its own location. We assume the transmission ranges of nodes are uniform. We modify Greedy Perimeter Stateless Routing(GPSR) to On-demand GPSR(OD-GPSR), which has several particular properties to meet our requirements.

First, OD-GPSR is a data-driven reactive routing protocol under which, only those nodes with data flowing over solicit location information from neighbors. As a result, unnecessary communication between neighbors is avoided to save the valuable energy.

Second, the destination in a packet is identified with location instead of node IP address. OD-GPSR uses home perimeter traversal method to identify the home node of the destination. Home perimeter is composed of all nodes surrounding a location in which home node is the nearest to the location. After taking a tour of the enclosed home perimeter of the target location and returning to the nearest node to the target location, a packet notices the loop and recognizes it as the home node of the target location.

Third, to maintain data consistency and improve robustness to node failures, OD-GPSR has the home node of a location recruit all neighbors as replica to cope with the dynamic property of sensor networks.

1. Soliciting Beacons from Neighbors

OD-GPSR is a reactive data-driven routing protocol and only those nodes over which data is flowing seek neighbor information for making routing decision. An example of the process to solicit beacons from neighbors is illustrated in Fig. 27. When a

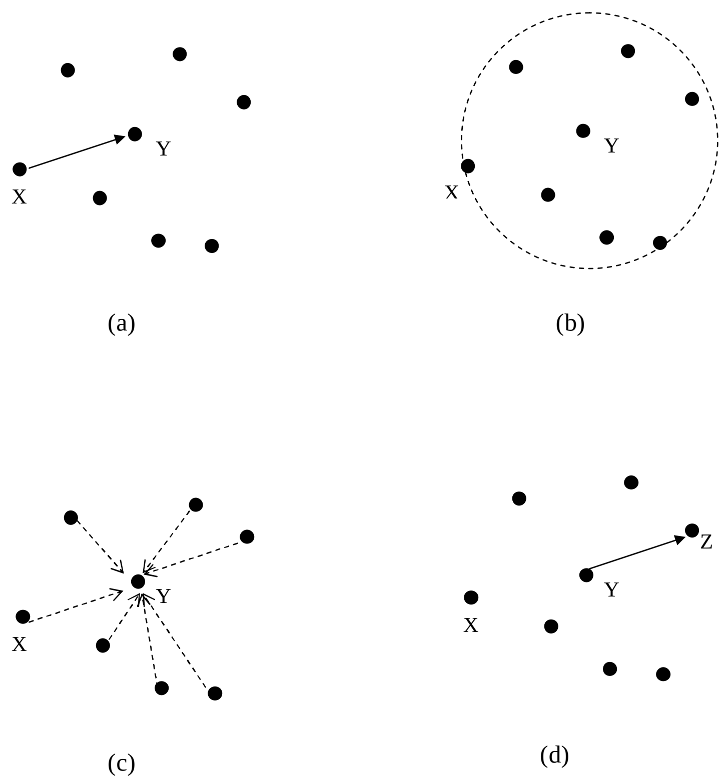

Fig. 27. The process for a node to solicit neighbor information

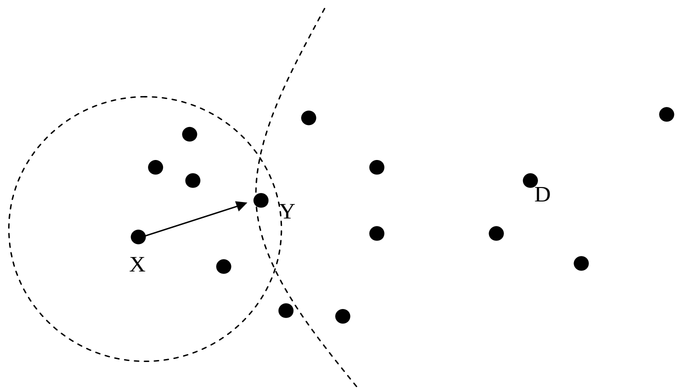

Fig. 28. Greedy forwarding example

node Y gets a packet to forward and finds itself with no reachability information of neighbors, the node broadcasts a beacon-request packet to its neighbors seeking location information, as shown in Fig. 27(b). In response, a neighbor node sends back a beacon including its location, as shown in Fig. 27(c). In Fig. 27(d), after collecting all neighbor location information, node Y makes a greedy forwarding to Z.

2. Greedy Forwarding

OD-GPSR is a localized algorithm and decisions of routing are based only on local neighbor information. Under OD-GPSR, each packet is marked by their originator with their target location and always starts with *greedy* mode. A forwarding node is always trying to make a greedy choice, selecting the neighbor geographically closest to the destination as the next hop. As shown in Fig. 28, node X has several neighbors including Y geographically closer to destination D. Note D is the *home node* for the destined location of packets from X. X selects Y as its next hop because Y is the one closest to the destined location. As a result, a packet is progressively moving closer to the destination.

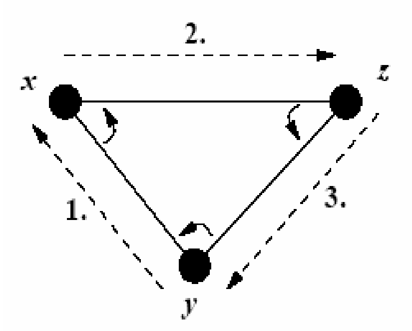

Fig. 29. Right-hand rule example

3. The Right-Hand Rule

Greedy forwarding fails when reaching a node which has no neighbors closer to the destination. Then the mode of the packet is changed to *perimeter forwarding*, in which the packet is forwarded using the *right-hand rule* to circumnavigate this region. The mode switches back to *greedy* when it reaches a node which is closer to the destination. Fig. 29 demonstrates the *right hand rule*. When arriving on an edge at node x, the packet is forwarded on the next edge counterclockwise about x from the ingress edge. This process lets the packet to tour enclosed faces as shown and then the packet circumnavigating regions where greedy forwarding fails. OD-GPSR routes perimeter forwarding mode packets on a planarized subgraph of the network connectivity graph, in which there are no crossing edges. A perimeter is a face of this planarized graph.

4. State Transformation of Nodes in OD-GPSR

Fig. 30 shows the state transformation of nodes in OD-GPSR, in which every nodes has three basic states. A node without neighbor information is in *standby* state; A node with neighbor information which can forward data without soliciting neighbor information is in *ready* state; A node being in the process of collecting neighbor information is in *time-out* state in which the node has no neighbor information for

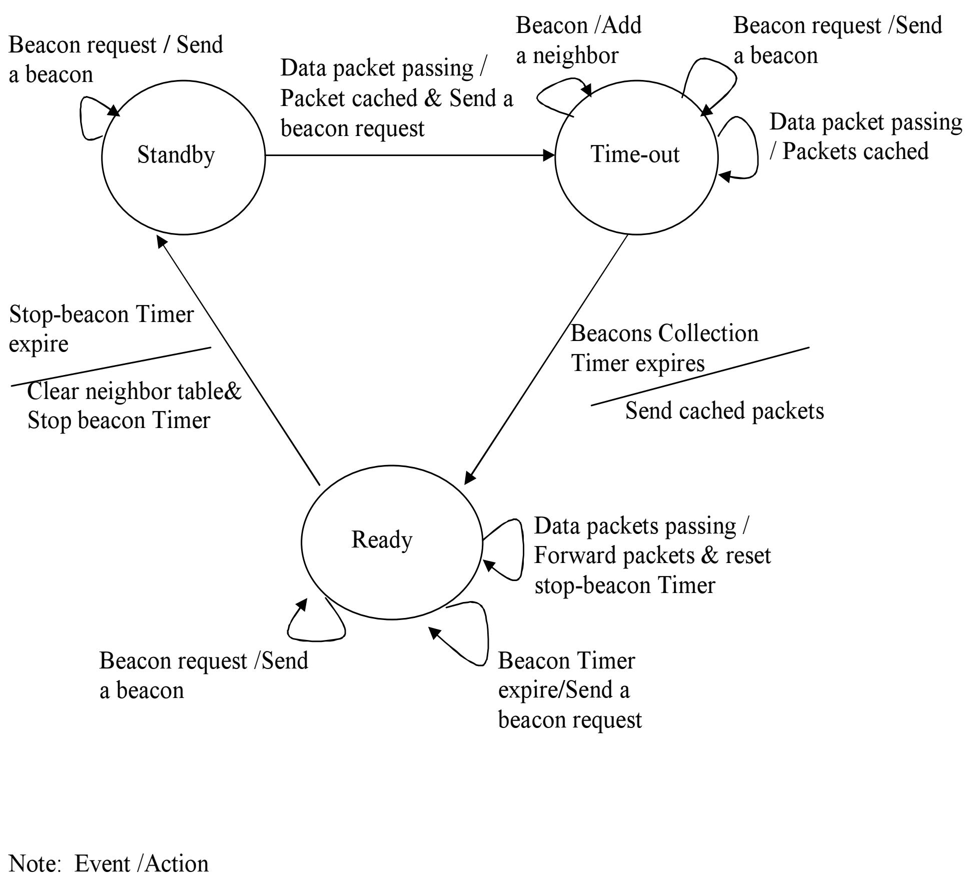

Fig. 30. Node state transformation

making routing decisions and any packets received have to be cached.

A node in standby state switches to time-out state after it broadcasts a beacon-request packet to neighbors when it originates a packet to send or has a packet to forward. The node then waits for a while to collect beacon packets from all neighbors which include location information. After believing having collected all neighbor information, the node switches to ready state and starts sending packets. In this state, the node periodically sends beacon-request packets to neighbors inquiring current state and updating the neighbor table upon receiving beacon packets. Each time a packet passing refreshes the ready state. After a certain period of time without data flowing over, the node stops sending beacon-request and all neighbor entries are still left in the neighbor table but the neighbor table is set with a time stamp. Thereafter, the node switches from ready state to standby state.

The reason not to remove all neighbor entries even when no traffic is flowing over is to prevent the *thrash* phenomenon in which a data traffic comes just after the removal of neighbor entries and thus incurs high delay because the node has to request neighbor information again. when a data traffic reaches a node with neighbor table, it checks the time stamp of the neighbor table and uses this neighbor information if the age of the neighbor table is within a time of period T beyond which the neighbor table is removed by the node.

Whenever a node receives beacon-request packet, it sends back a beacon packet with its location information regardless of its current state. To avoid synchronization of neighbors' beacons, we jitter beacon's transmissions by a period of time of the interval between beacon-request packets.

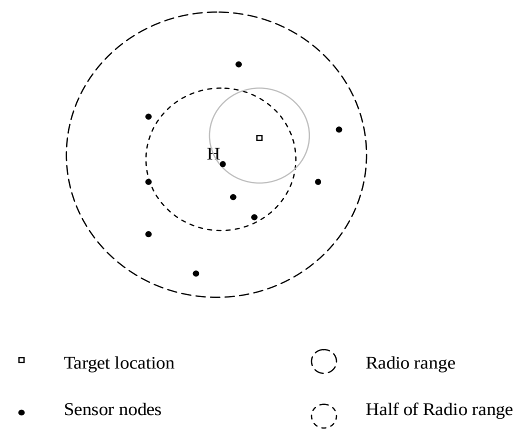

Fig. 31. Identify the target transient home node

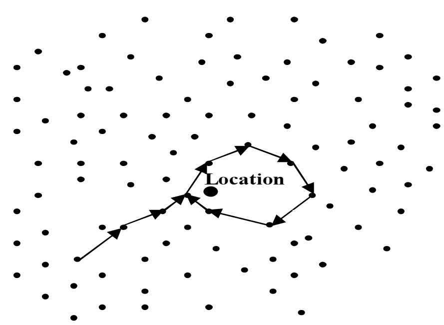

Fig. 32. An example of a packet to find the home node of its target location

5. Data Consistency Problem

With the destination marked with location, a packet reaches the home node of the destination. Home nodes are categorized into two types: *transient home node* and *persistent home node*. For transient home node, it does not matter if the same data can be retrieved from the same location in different time. In contrast, data consistency is required for persistent home node, meaning that the same data sent to a location before should be retrieved later from the home node of the same location regardless the possible changes of the home node for the location.

A transient home node is identified when a packet reaches a node whose distance from the target location is less than half of its radio range and no neighbor nodes are closer to the destination, as shown in Fig. 31. The other way to identify a transient home node is the perimeter traversal method in [39] as shown in Fig. 32. After the packet returns to the nearest node to the destined location and finds itself traversing a loop, the node is recognized as the home node for the location.

For persistent home node, the first time a packet is sent to a location, OD-GPSR identifies the target home node using the perimeter traversal method. Following packets to the same destination reach destination as they arrive at the marked home node for the target location without the traversal of the perimeter. Due to the dynamic nature of sensor networks, OD-GPSR has special mechanism for persistent home node to keep the data consistency. The first time a node is identified and marked as a persistent home node for a location, it recruits all neighbors as replica nodes. Each replica node has a timer associated with it. The primary home node broadcasts refresh packets periodically to refresh timers on all neighbors. When the home node is dead, the timer in a replica node will expire and the replica node will keep sending a special packet to the target location reporting the death of the primary home node until receiving response. To handle the problem of new emerging home node for persistent home node, the current primary home node sends packets to the target location periodically to check the existence of new home node.

C. Simulation

We implement the design of the modified version of GPSR proposed in [39](We use the term GPSR to refer this modified version of GPSR.) in order to compare its performance with OD-GPSR.

We simulated OD-GPSR in ns-2 [34]. We use the CMU wireless extensions which includes full simulation of the IEEE 802.11 physical and MAC layer. Our simulations are for networks of 256 nodes with 802.11 WaveLAN radios. The radio range is changed to 40 meters to make it closer to the real situation. The nodes are randomly deployed in a 256m by 256 m rectangle area.

Three metrics are used for evaluation of performance. **Average energy consumption** is defined as the ratio of the total dissipated energy per node in the network to the number of packets successfully received by the sink. This metric defines the energy efficiency of the protocol. **Packet delivery success rate** is defined as the ratio of the number of data packets successfully received by the sink to the number of data packets sent by the data sources. This metric defines the effectiveness of data delivery. **Average delay** is defined as the average time between the moment a data packet is sent by a data source and the moment the sink receives the data packet. This metric defines the freshness of data packets.

Next we show simulation results to demonstrate the performance of OD-GPSR in comparison to GPSR. Since the latter works under the assumption of known boundary, we limit traffic destinations inside the network topology for the convenience of comparison. Error model is introduced in the simulation to make it close to the real situation, in which 10 percent randomly selected nodes keep switching between disabled state and normal state with intervals of 100 seconds. Simulations with various beacon interval parameter are run for both GPSR and OD-GPSR implementations to study the performance. Note that in the following figures, GPSR-bint5 means GPSR with beacon interval of 5 seconds and ODGPSR-bint5 means OD-GPSR with beacon interval of 5 seconds, and so on. Fig. 33, Fig. 34, and Fig. 35 show results of this simulation.

From Fig. 33, ODGPSR-bint10 expends less energy than GPSR-bint10 when the

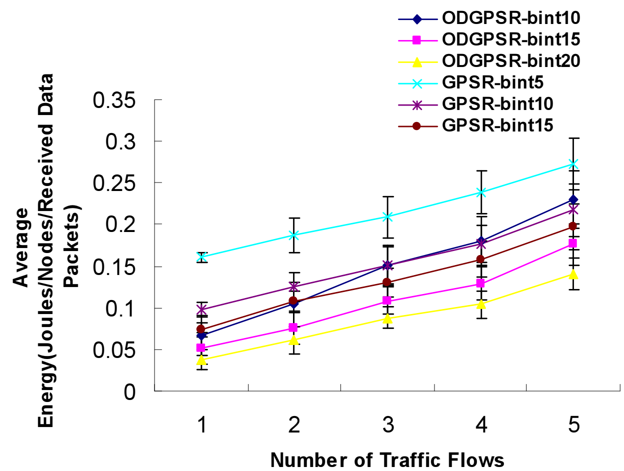

Fig. 33. Average energy consumption of GPSR simulation

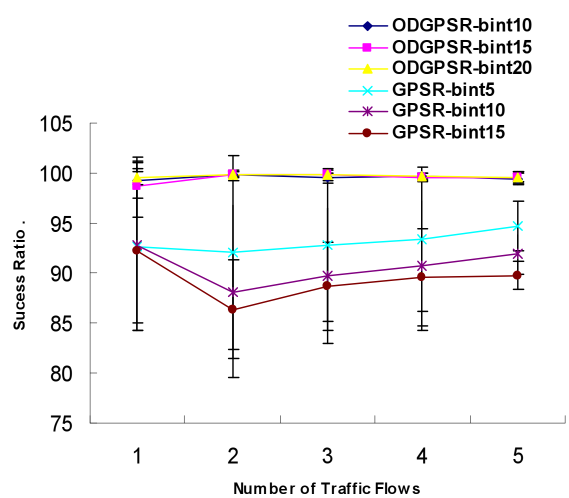

Fig. 34. Packets delivery success rate of GPSR simulation

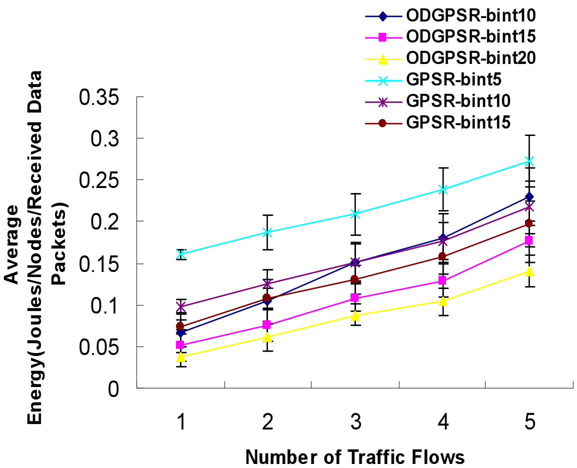

Fig. 35. Packets average delay of GPSR simulation

traffic is less, although the delivery rate of ODGPSR-bint10 is higher than GPSR-bint10 as shown in Fig. 34. The difference of the energy consumption is incurred by those unnecessary communications among nodes in GPSR. This result is also demonstrated by comparing ODGPSR-bint15 and GPSR-bint15. In OD-GPSR, only nodes with traffic solicit neighbor information and thus saves energy by avoiding unnecessary communications. Imagine a large-scale sensor network with not much traffic at most time, the energy saving is significant. As traffic involves more nodes, the saving becomes less as shown in Fig. 33, in which, when the number of traffic flows increases, the difference between GPSR-bint10 and ODGPSR-bint10 decreases. In OD-GPSR, neighbor information is required to be collected within the beacon interval upon receiving request, which intends to incur high traffic collisions when traffic is high. This is a disadvantage of OD-GPSR than GPSR in which each node randomly sends beacons to neighbors periodically with better desynchronization effect. Therefore, a small value of beacon period is not good for OD-GPSR. As shown in Fig. 33, when the number of traffic is high, the ODGPSR-bint10 is not better than GPSR-bint10 in terms of energy efficiency, while ODGPSR-bint15 is still much better than GPSR-bint15.

As shown in Fig. 34, the delivery rate of OD-GPSR is much better than GPSR. The reason is that GPSR always uses the home perimeter traversal algorithm for recognition of the home node, while OD-GPSR makes improvements for both types of home nodes and therefore the home node of the target location could be determined earlier in many cases especially when the density of nodes is relatively high.

As shown in Fig. 35, the average delay for packet delivery in OD-GPSR is higher than that in GPSR. The delay in OD-GPSR is caused by the delay of the first several packets in each traffic flow. When packets reach nodes not visited recently, they are cached in the nodes until the nodes get neighbor information. When a node already has the neighbor information, following packets are forwarded immediately resulting in comparable delay with GPSR. Under OD-GPSR, the average delay will drop correspondingly as the number of consecutive packets in a flow increases. So we can conclude that the average packet delivery delay of OD-GPSR is a little higher than(but very close to) that of GPSR when the number of consecutive packets in a flow is sufficiently large.

D. Contributions

In order to apply geographical routing protocol Greedy Perimeter Stateless Routing (GPSR) in wireless sensor networks to support our shemes, several problems are fixed. First, packet destinations are marked with locations instead of identifiers like IP address and packets are routed to the home node of the destination. Second, the on-demand nature saves energy by elimination of unnecessary communication among sensor nodes. Third, data consistency problem is solved.

This new version of GPSR is called On-demand GPSR, which is a data driven geographical routing protocol customized for sensor networks. OD-GPSR adapts to the unique requirements for applications in sensor networks and therefore can be better applied in sensor networks. Simulation results show that OD-GPSR performs well in terms of energy efficiency and packet delivery rate at the cost of a little bit more packet delivery delay.

CHAPTER VI

CONCLUSION AND FUTURE WORK

Energy efficient data dissemination is one of the most important topics in sensor networks. Based on the study of this problem, we propose two efficient data dissemination schemes for two distinct categories of applications in large-scale wireless sensor networks. The first scheme is for applications like temperature monitoring, in which sinks periodically retrieve data from the network. The second scheme is for application such as habitat monitoring in which sensor networks are designed to monitor mobile objects like animals. Our methods improve the performance in energy efficiency in following features. First, minimize flooding for sending queries. Second, consider data redundancy and decrease transmission of redundant data to save valuable energy. Third, reduce the transmission of data for energy efficiency by providing multi-resolution data dissemination for applications which accept data with less detail under some circumstance. Our methods are best suitable for large-scale sensor networks with high node density.

In the design of the schemes, we did not take terrain effect on the detection into account when we propose the spatial-based multi-resolution data model. However, a spatial-based multi-resolution data model taking account the terrain effect would be more real in practice. Correspondingly, new data dissemination schemes are required based on the new data model. We take this as future works.

REFERENCES

[1] L. Barri‘ere, P. Fraigniaud, and L. Narayanan, "Robust Position-Based Routing in Wireless Ad Hoc Networks with Unstable Transmission Ranges," in *Proc. of the 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications(DIAL-M)*, Rome, Italy, pp. 19–27, 2001.

[2] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Wireless Networks*, vol. 7, pp. 609–616, 2001.

[3] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000.

[4] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Network Topologies," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, July 2004.

[5] J. Chen, Y. Guan, and U. Pooch, "A Spatial-based Multi-resolution Data Dissemination Scheme for Wireless Sensor Networks," in *Proc. of the 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (IEEE WMAN 05),* Denver, CO, pp. 212–219, April, 2005.

[6] J. Chen, Y. Guan, and U. Pooch, "Customizing a Geographical Routing Protocol for Wireless Sensor Networks," in *Proc. of the International Conference on Information Technology(ITCC 2005),* Las Vegas, NV, pp. 185–190, April, 2005.

[7] J. Chen, Y. Guan, and U. Pooch, "Customizing GPSR for Wireless Sensor Networks," in *Proc. of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems(MASS 2004),* Fort Lauderdale, FL, pp. 549–551, October, 2004.

[8] J. Chen, Y. Guan, and U. Pooch, "An Efficient Data Dissemination Method in Wireless Sensor Networks," in *Proc. of the IEEE Global Telecommunications Conference, Globecom 2004,* Dallas, TX, pp. 145–150, November, 2004.

[9] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," in *Proc. of the 7th ACM International Conference on Mobile Computing and Networking*, Rome, Italy, pp. 85–96, July, 2001.

[10] D. A. Coffin, D. J. Van Hook, S. M. McGarry, and S. R. Kolek, "Declarative Ad Hoc Sensor Networking," in *Proc. of SPIE Integrated Command Environments Conference*, San Diego, CA, pp. 109–120, July 2000.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, Cambridge, MA, MIT Press, 2001, pp. 570–573.

[12] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in *Proc. of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99)*, Seattle, WA, pp. 263–270, August 1999.

[13] Q. Fang, J. Gao, and L. J. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," in *Proc. of the 23th International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, Hong Kong, pp. 201–211, March, 2004.

[14] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks," Technical Report UCLA/CSD-TR-02-0013, Computer Science Department, UCLA, Los Angles, CA, 2002.

[15] D. Ganesan, S. Ratnasamy, H. Wang, and D. Estrin, "Coping with Irregular Spatio-temporal Sampling in Sensor Networks," in *Proc. of the Second Workshop on Hot Topics In Networks (HotNets-II)*, Cambridge, MA, pp. 125–130, June, 2003.

[16] D. Ganesan, B. Greenstein. D. Perelyubskiy, D. Estrin and J. Heidemann, "An Evaluation of Multi-resolution Search and Storage in Resource-constrained Sensor Networks," in *Proc. of the First ACM Conference on Embedded Networked Sensor Systems(SenSys 2003)*, Los Angeles, CA, pp. 89–102, November 2003.

[17] J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Geometric Spanner for Routing in Mobile Networks," in *Proc. of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing(MobiHoc 2001)*, Long Beach, CA, pp. 45–55, October 2001.

[18] L. Girod and D. Estrin, "Robust Range Estimation Using Acoustic, and Multimodal Sensing," in *Proc. of the IEEE Conference on Intelligent Robots and Systems*, Maui, HI, pp. 1–9, October, 2001.

[19] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker, "DIFS: A Distributed Index for Features in Sensor Networks," in *Proc. of 1st IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, pp. 201–211, May 2003.

[20] H. Gupta, S. R. Das, and Q. Gu, "Connected Sensor Cover: Self-Organization of the Sensor Network for Efficient Query Execution," in *Proc. of the Fourth Annual ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Annapolis, MD, pp. 189–200, June 2003.

[21] P. S. Heckbert and M. Garland, "Multiresolution Modeling for Fast Rendering," *Graphics Interface '94*, Banff, Canada, pp. 43–50, May 1994.

[22] J. Heidemann and N. Bulusu, "Using Geospatial Information in Sensor Networks," in *Proc. of CSTB Workshop on Intersection of Geospatial Information and Information Technology*, Arlington, VA, pp. 61-65, October 2001.

[23] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building Efficient Wireless Sensor Networks with Low-level Naming," in *Proc. of the Symposium on Operating Systems Principles*, Banff, Alberta, Canada, pp. 146–159, October 2001.

[24] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *Proc. of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 1999)*, Seattle, WA, pp. 174–185, August 1999.

[25] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000)*, Boston, MA, pp. 56–67, 2000.

[26] B. Karp, and H. T. Kung, "Greedy Perimeter Stateless Routing," in *Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000)*, Boston, MA, pp. 243–254, 2000.

[27] Y. Ko, and N. Vaidya, "Location-aided Routing in Mobile Ad Hoc Networks," in *Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 98)*, Dallas, TX, pp. 66–75, August 1998.

[28] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing," in *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Annapolis, MD, pp. 267–278, June 2003.

[29] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad-Hoc Networks Beyond Unit Disk Graphs," in *Proc. of the 1st ACM DIALM-POMC Joint Workshop on Foundations of Mobile Computing(DIALM-POMC)*, San Diego, CA, pp. 69–78, September 2003.

[30] J. Li, J. Jannotti, D. Couto, D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," in *Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000)*, Boston, MA, pp. 120–130, 2000.

[31] X. Li, Y. J Kim, R. Govindan, and W. Hong, "Multi-dimensional Range Queries in Sensor Networks," in *Proc. of the First ACM Conference on Embedded Networked Sensor Systems(SenSys 2003)*, Los Angeles, CA, pp. 63–75, November 2003.

[32] W. Liao, J. Sheu, and Y. Tseng, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," *Telecommunication Systems*, vol. 18, pp. 37–60, 2001.

[33] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks," in *Proc. of the 5th Symposium on Operating System Design and Implementation Special Issue: Physical Interface (OSDI 2002)*, Boston, MA, pp. 131–146, 2002.

[34] S. McCanne, and S. Floyd, *ns Network Simulator*, http://www.isi.edu/nsnam/ns/, 2005.

[35] G. Pottie, and W. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, pp. 51–58, May 2000.

[36] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location Sup-

port System," in *Proc. of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000)*, Boston, MA, pp. 32–43, 2000.

[37] N. Priyantha, A. Liu, H. Balakrishnan and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," in *Proc. of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2001)*, Rome, Italy, pp. 1–14, 2001.

[38] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," in *Proc. of the 19th International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, Tel-Aviv, Israel, pp. 404–413, March 2000.

[39] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin and F. Yu, "Data-centric Storage in Sensornets with GHT, a Geographic Hash Table," *Mobile Networks and Applications*, vol. 8, pp. 427–442, August 2003.

[40] A. Savvides, C.-C. Han, and M. B. Srivastava, "Dynamic Fine-Grain Localization in Ad-Hoc Networks of Sensors," in *Proc. of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking(Mobicom 2001)*, Rome, Italy, pp. 166–179, 2001.

[41] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing Sensor Networks in the Energy-Latency-Density Design Space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, January 2002.

[42] S. Woo, and S. Singh, "Scalable Routing Protocol for Ad Hoc Networks," *Journal of Wireless Networks (WINET)*, vol. 7, pp. 513-529, September 2001.

[43] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Re-

liable Multihop Routing in Sensor Networks," in *Proc. of the 1st International Conference on Embedded Networked Sensor Systems (ACM SenSys 2003)*, Los Angeles, CA, pp. 14–27, November 2003.

[44] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," in *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, pp. 70–84, July 2001.

[45] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks," in *Proc. of the Eighth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2002)*, Atlanta, GA, pp. 148–159, September 2002.

[46] Y. Yu, R. Govindan, and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," Technical Report UCLA/CSD-TR-01-0023, Computer Science Department, UCLA, Los Angles, CA, May 2001.

[47] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of Radio Asymmetry on Wireless Sensor Networks," in *Proc. of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys'04)*, Boston, MA, pp. 125-138, June 2004.

APPENDIX A

SIMULATION DATA OF THE PROACTIVE DATA DISSEMINATION SCHEME

The data included in this appendix is of the proactive spatial-based multi-resolution data dissemination scheme described in Chapter III. The simulation is run on ns-2 to evaluate the performance of this scheme in comparison with Directed Diffusion. Three metrics are used for the evaluation: packet delivery success rate, average delay and average energy consumption.

For SMRDD, different values of the grid size for data dissemination in the highest resolution level are used for simulations to evaluate the performance for applications with various types of sensor nodes. These sensor nodes may have different sensing ranges which determine the grid sizes corresponding to the highest data resolution level in SMRDD.

Table I. SMRDD Grid Size = 16m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.19%	0.188209	0.0009884
99.09%	0.397618	0.0016265
99.57%	0.179465	0.0009995
98.79%	0.225263	0.0011753
99.51%	0.186814	0.0011513
99.38%	0.152635	0.0011548

Table II. SMRDD Grid Size = 22m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.61%	0.133974	0.0009944
99.37%	0.438046	0.0017164
99.72%	0.136959	0.0009780
98.33%	0.178825	0.0011727
99.83%	0.130065	0.0010169
99.64%	0.175078	0.0011933

Table III. SMRDD Grid Size = 32m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.52%	0.514140	0.0020948
99.93%	0.190322	0.0012846
99.44%	0.232272	0.0013722
98.98%	0.156997	0.0011913
99.91%	0.160989	0.0012258
99.98%	0.200900	0.0012267

Table IV. SMRDD Grid Size = 48m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.156792	0.0010141
99.86%	0.502126	0.0020780
99.85%	0.239373	0.0013473
98.90%	0.239313	0.0007759
99.97%	0.410859	0.0021809
99.97%	0.214675	0.0016174

Table V. SMRDD Grid Size = 64m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.773419	0.0003050
100%	0.452220	0.0020586
99.45%	0.702900	0.0032077
98.94%	0.463092	0.0024005
100%	0.446158	0.0024018
99.87%	0.483776	0.0028505

Table VI. Simulation Results of Directed Diffusion

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
48.79%	1.911182	0.0137235
49.91%	2.113621	0.0138523
50.01%	1.735691	0.0157221
47.99%	1.847104	0.0136035
46.67%	1.503972	0.0147274
51.06%	1.497625	0.0138125

Table VII. Averaged Data of Three Metrics for Directed Diffusion

	Average	Standard Deviation
Packet Delivery Success Rate	48.95%	1.53%
Average Delay	1.768199	0.240861
Average Energy Consumption	0.0142405	0.0008291

Table VIII. Simulation Results of SMRDD: Packet Delivery Success Rate

Grid side length of SMRDD	Packet Delivery Success Rate	Standard Deviation
16	93.26%	0.29%
22	93.58%	0.20%
32	95.79%	0.25%
48	99.93%	0.06%
64	99.88%	0.22%

Table IX. Simulation Results of SMRDD: Packet Delivery Average Delay

Grid side length of SMRDD	Average Delay	Standard Deviation
16	0.221667	0.089277
22	0.198825	0.119123
32	0.242603	0.135868
48	0.293856	0.132714
64	0.553594	0.145258

Table X. Simulation Results of SMRDD: Average Energy Consumption

Grid side length of SMRDD	Average Energy Consumption	Standard Deviation
16	0.001183	0.0002324
22	0.001179	0.0002792
32	0.001399	0.0003467
48	0.001502	0.0005648
64	0.002204	0.0010129

APPENDIX B

SIMULATION DATA OF THE REACTIVE DATA DISSEMINATION SCHEME

The data included in this appendix is of the reactive multi-resolution data dissemination scheme described in Chapter IV. The simulation is run on ns-2 to evaluate the performance of this scheme in comparison with Directed Diffusion. Three metrics are used for the evaluation: packet delivery success rate, average delay and average energy consumption.

For SMRDD, different values of the sensing radius for data dissemination in the highest resolution level are used for simulations to evaluate the performance for applications with various types of sensor nodes.

Table XI. SMRDD Sensing Radius = 0m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
94.33%	0.340	0.0161
96.23%	0.310	0.0201
96.34%	0.430	0.0154
90.12%	0.330	0.0179
95.07%	0.470	0.0173
94.79%	0.410	0.0189

Table XII. SMRDD Sensing Radius = 10m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
93.45%	0.350	0.0143
95.31%	0.330	0.0156
92.04%	0.410	0.0135
95.38%	0.390	0.0153
94.79%	0.320	0.0144
94.77%	0.350	0.0139

Table XIII. SMRDD Sensing Radius = 20m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
98.34%	0.210	0.0075
99.01%	0.230	0.0089
97.59%	0.190	0.0067
99.23%	0.290	0.0101
97.99%	0.310	0.0061
96.87%	0.190	0.0071

Table XIV. SMRDD Sensing Radius = 30m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
98.35%	0.177	0.0059
98.76%	0.169	0.0081
97.13%	0.221	0.0055
99.01%	0.189	0.0049
98.79%	0.193	0.0047
97.52%	0.197	0.0048

Table XV. SMRDD Sensing Radius = 40m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.15%	0.141	0.0053
98.73%	0.152	0.0061
99.78%	0.134	0.0055
97.91%	0.147	0.0047
99.56%	0.149	0.0052
99.38%	0.139	0.0049

Table XVI. SMRDD Sensing Radius = 50m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.29%	0.161	0.0052
99.75%	0.159	0.0057
98.37%	0.153	0.0067
98.75%	0.173	0.0045
99.34%	0.167	0.0043
98.73%	0.165	0.0051

Table XVII. SMRDD Sensing Radius = 60m

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
98.42%	0.170	0.0040
98.56%	0.130	0.0037
97.61%	0.250	0.0039
99.42%	0.210	0.0042
99.03%	0.110	0.0058
99.19%	0.120	0.0055

Table XVIII. Directed Diffusion

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
68.23%	1.880	0.1367
74.82%	1.930	0.1232
71.93%	1.360	0.1179
69.41%	2.010	0.1056
75.19%	2.130	0.1429
70.09%	1.620	0.1511

Table XIX. Directed Diffusion: Averaged Data of Three Metrics

	Average	Standard Deviation
Packet Delivery Success Rate	71.61%	2.89%
Average Delay	1.8217	0.2826
Average Energy Consumption	0.1295	0.01699

Table XX. SMRDD: Packet Delivery Success Rate

Sensing Radius of SMRDD	Packet Delivery Success Rate	Standard Deviation
0	94.48%	2.28%
10	94.29%	1.30%
20	98.17%	0.89%
30	98.26%	0.76%
40	99.08%	0.68%
50	99.03%	0.51%
60	98.71%	0.66%

Table XXI. SMRDD: Packet Delivery Average Delay

Grid side length of SMRDD	Average Delay	Standard Deviation
0	0.380	0.06400
10	0.358	0.03488
20	0.237	0.05160
30	0.191	0.01802
40	0.143	0.00680
50	0.163	0.00690
60	0.165	0.05576

Table XXII. SMRDD: Average Energy Consumption

Grid side length of SMRDD	Average Energy Consumption	Standard Deviation
0	0.0176	0.00174
10	0.0145	0.00081
20	0.0077	0.00149
30	0.0057	0.00128
40	0.0053	0.00049
50	0.0052	0.00087
60	0.0045	0.00089

APPENDIX C

SIMULATION DATA OF OD-GPSR

The data included in this appendix is of On Demand-GPSR described in Chapter V. The simulation is run on ns-2 to evaluate the performance of this protocol in comparison with GPSR. Three metrics are used for the evaluation: packet delivery success rate, average delay and average energy consumption.

Note that in the following tables, GPSR-bint5 means GPSR with beacon interval of 5 seconds and ODGPSR-bint5 means OD-GPSR with beacon interval of 5 seconds, and so on.

Table XXIII. GPSR: 1 Flow, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
96.71%	0.051082	0.077
100%	0.045652	0.068
91.13%	0.045462	0.071
92.61%	0.061209	0.085
77.34%	0.049097	0.075
100%	0.040849	0.072

Table XXIV. GPSR: 2 Flows, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
85.40%	0.049123	0.110
98.27%	0.034710	0.087
77.97%	0.073327	0.099
84.16%	0.044406	0.111
75.00%	0.056839	0.145
97.28%	0.040770	0.092

Table XXV. GPSR: 3 Flows, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
87.81%	0.055605	0.148
99.01%	0.031211	0.101
81.05%	0.044340	0.128
85.67%	0.042051	0.131
81.71%	0.047674	0.167
97.03%	0.033359	0.105

Table XXVI. GPSR: 4 Flows, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
90.58%	0.034479	0.143
92.81%	0.032884	0.129
93.56%	0.043050	0.160
90.95%	0.038216	0.142
94.30%	0.040000	0.183
75.09%	0.057027	0.194

Table XXVII. GPSR: 5 Flows, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
89.89%	0.055297	0.215
90.29%	0.038592	0.183
88.01%	0.044841	0.211
93.06%	0.032003	0.155
91.18%	0.046962	0.241
86.03%	0.037646	0.180

Table XXVIII. GPSR: 1 Flow, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.045667	0.089
92.12%	0.045643	0.091
97.54%	0.061048	0.109
83.25%	0.049955	0.095
83.55%	0.057196	0.108
100%	0.040796	0.093

Table XXIX. GPSR: 2 Flows, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
89.36%	0.048084	0.124
98.76%	0.034757	0.108
81.19%	0.042895	0.125
85.15%	0.044285	0.129
77.23%	0.057260	0.157
97.03%	0.040722	0.113

Table XXX. GPSR: 3 Flows, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
88.96%	0.054795	0.165
99.01%	0.031228	0.123
82.70%	0.045022	0.153
88.14%	0.041816	0.151
82.87%	0.047978	0.181
96.38%	0.033515	0.128

Table XXXI. GPSR: 4 Flows, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
78.19%	0.056593	0.206
92.81%	0.032380	0.150
94.18%	0.043056	0.180
91.95%	0.038008	0.160
96.28%	0.048422	0.201
90.95%	0.034530	0.165

Table XXXII. GPSR: 5 Flows, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
92.07%	0.055424	0.234
92.57%	0.038471	0.195
90.29%	0.045380	0.230
93.56%	0.032213	0.175
93.95%	0.047289	0.263
88.60%	0.038612	0.206

Table XXXIII. GPSR: 1 Flow, bint = 5

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.045823	0.152
91.63%	0.046090	0.157
97.04%	0.061622	0.175
87.68%	0.050956	0.157
79.80%	0.056095	0.166
100%	0.040933	0.156

Table XXXIV. GPSR: 2 Flows, bint = 5

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
91.34%	0.048015	0.184
98.51%	0.034804	0.175
86.63%	0.045073	0.191
91.09%	0.044506	0.184
87.38%	0.057797	0.209
97.28%	0.040524	0.176

Table XXXV. GPSR: 3 Flows, bint = 5

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
91.60%	0.054404	0.221
99.01%	0.031288	0.189
86.66%	0.054812	0.217
91.93%	0.041888	0.205
90.61%	0.049402	0.231
96.54%	0.033568	0.190

Table XXXVI. GPSR: 4 Flows, bint = 5

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
83.89%	0.056155	0.261
94.67%	0.032519	0.211
95.29%	0.042864	0.244
95.66%	0.038820	0.224
97.65%	0.049044	0.265
92.94%	0.034980	0.226

Table XXXVII. GPSR: 5 Flows, bint = 5

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
95.34%	0.054592	0.288
95.14%	0.037885	0.249
93.66%	0.045943	0.284
95.04%	0.032977	0.238
96.43%	0.047592	0.314
92.47%	0.039715	0.265

Table XXXVIII. OD-GPSR: 1 Flow, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.071969	0.032
100%	0.158956	0.056
100%	0.146821	0.061
100%	0.135933	0.069
95.57%	0.382044	0.099
100%	0.096416	0.085

Table XXXIX. OD-GPSR: 2 Flows, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.078680	0.066
100%	0.081687	0.077
100%	0.090839	0.116
100%	0.097967	0.126
99.50%	0.155147	0.107
100%	0.094538	0.134

Table XL. OD-GPSR: 3 Flows, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.34%	0.119160	0.120
98.85%	0.146429	0.132
100%	0.121629	0.178
100%	0.088824	0.165
100%	0.089165	0.141
99.01%	0.133139	0.170

Table XLI. OD-GPSR: 4 Flows, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.50%	0.108065	0.161
100%	0.087278	0.149
99.75%	0.101735	0.198
99.26%	0.128183	0.214
100%	0.084081	0.151
100%	0.110764	0.204

Table XLII. OD-GPSR: 5 Flows, bint = 10

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.70%	0.123166	0.235
99.80%	0.139672	0.244
99.80%	0.100918	0.243
98.01%	0.132328	0.233
99.50%	0.123712	0.189
98.32%	0.222354	0.296

Table XLIII. OD-GPSR: 1 Flow, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.078662	0.022
99.51%	0.219384	0.050
100%	0.146604	0.043
99.51%	0.140789	0.052
92.61%	0.384389	0.078
100%	0.099460	0.061

Table XLIV. OD-GPSR: 2 Flows, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.081465	0.047
100%	0.082827	0.055
100%	0.089810	0.081
100%	0.100471	0.090
98.76%	0.146759	0.086
100%	0.094934	0.096

Table XLV. OD-GPSR: 3 Flows, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.099605	0.086
99.67%	0.159428	0.095
99.67%	0.138167	0.127
100%	0.088881	0.119
100%	0.090616	0.102
99.51%	0.121593	0.122

Table XLVI. OD-GPSR: 4 Flows, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.098383	0.117
99.88%	0.088667	0.109
98.76%	0.125609	0.142
99.50%	0.120115	0.154
100%	0.082171	0.109
99.50%	0.105414	0.148

Table XLVII. OD-GPSR: 5 Flows, bint = 15

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.451821	0.171
99.80%	0.162700	0.185
100%	0.101788	0.174
99.31%	0.145851	0.180
99.70%	0.262744	0.136
98.61%	0.174754	0.212

Table XLVIII. OD-GPSR: 1 Flow, bint = 20

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.078707	0.018
99.51%	0.179086	0.036
100%	0.155840	0.035
100%	0.188987	0.042
98.03%	0.332541	0.048
100%	0.095938	0.048

Table XLIX. OD-GPSR: 2 Flows, bint = 20

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.076809	0.038
100%	0.082474	0.044
100%	0.092084	0.065
100%	0.095875	0.072
99.26%	0.202421	0.069
100%	0.094835	0.076

Table L. OD-GPSR: 3 Flows, bint = 20

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
100%	0.099714	0.069
98.85%	0.169381	0.083
99.84%	0.130272	0.103
100%	0.088507	0.095
100%	0.089945	0.082
100%	0.114058	0.097

Table LI. OD-GPSR: 4 Flows, bint = 20

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.75%	0.104379	0.094
100%	0.086779	0.088
99.75%	0.100677	0.113
99.01%	0.134721	0.125
100%	0.084304	0.088
99.63%	0.110797	0.117

Table LII. OD-GPSR: 5 Flows, bint = 20

Packet Delivery Success Rate	Average Delay	Average Energy Consumption
99.80%	0.117121	0.135
99.40%	0.134015	0.143
100%	0.103873	0.140
99.70%	0.151228	0.147
99.90%	0.122187	0.113
99.61%	0.177641	0.170

Table LIII. OD-GPSR: Packet Delivery Success Rate

Flows	OD-bint10	STDEV	OD-bint15	STDEV	OD-bint20	STDEV
1	99.26%	1.81%	98.61%	2.95%	99.59%	0.79%
2	99.92%	0.20%	99.79%	0.51%	99.87%	0.30%
3	99.53%	0.54%	99.81%	0.22%	99.78%	0.46%
4	99.75%	0.31%	99.60%	0.47%	99.69%	0.36%
5	99.36%	0.59%	99.57%	0.53%	99.57%	0.51%

Table LIV. GPSR: Packet Delivery Success Rate

Flows	GPSR-bint5	STDEV	GPSR-bint10	STDEV	GPSR-bint15	STDEV
1	92.69%	8.49%	92.74%	7.78%	92.22%	7.97%
2	92.04%	9.66%	88.12%	8.59%	86.35%	4.93%
3	92.73%	7.66%	89.68%	6.78%	88.72%	4.41%
4	93.35%	7.23%	90.73%	6.41%	89.55%	4.87%
5	94.68%	2.46%	91.84%	2.04%	89.74%	1.39%

Table LV. OD-GPSR: Packet Delivery Average Delay

Flows	OD-bint10	STDEV	OD-bint15	STDEV	OD-bint20	STDEV
1	0.16500	0.111066	0.1782	0.111981	0.17185	0.090402
2	0.09980	0.028103	0.0994	0.024299	0.10742	0.047141
3	0.11639	0.023320	0.1164	0.028483	0.11530	0.030835
4	0.10335	0.016286	0.1034	0.017144	0.10360	0.018356
5	0.14030	0.042229	0.2166	0.126729	0.13430	0.026581

Table LVI. GPSR: Packet Delivery Average Delay

Flows	GPSR-bint5	STDEV	GPSR-bint10	STDEV	GPSR-bint15	STDEV
1	0.050253	0.006979	0.050051	0.007698	0.048454	0.007587
2	0.045120	0.013725	0.044667	0.007580	0.049862	0.007707
3	0.044227	0.009088	0.042392	0.008896	0.042373	0.010285
4	0.042397	0.008697	0.042165	0.009159	0.040943	0.008934
5	0.043117	0.008224	0.042898	0.008179	0.042557	0.007765

Table LVII. OD-GPSR: Average Energy Consumption

Flows	OD-bint10	STDEV	OD-bint15	STDEV	OD-bint20	STDEV
1	0.067	0.02338	0.0510	0.01863	0.0378	0.01122
2	0.104	0.02724	0.0758	0.02001	0.0610	0.01577
3	0.151	0.02327	0.1085	0.01653	0.0882	0.01247
4	0.179	0.02905	0.1295	0.02047	0.1040	0.01614
5	0.230	0.03416	0.1760	0.02458	0.1410	0.01845

Table LVIII. GPSR: Average Energy Consumption

Flows	GPSR-bint5	STDEV	GPSR-bint10	STDEV	GPSR-bint15	STDEV
1	0.1605	0.00596	0.0975	0.00876	0.0742	0.00846
2	0.1865	0.02077	0.1260	0.01714	0.1073	0.01250
3	0.2088	0.02516	0.1510	0.02195	0.1300	0.01714
4	0.2385	0.02548	0.1770	0.02275	0.1585	0.02173
5	0.2730	0.03067	0.2172	0.03144	0.1975	0.02790

VITA

Jian Chen was born in Shandong Province, PR China. In 1988, he entered Qingdao University and received his B.E. degree in computer engineering in 1992. In 1997, he entered Peking University and received his M.S. degree in computer science in 2000. He started pursuing a Ph.D. degree in computer science at Texas A&M University in 2000. Since then, he has worked as a graduate research assistant and teaching assistant in Computer Science Department.

Following is his permanent address: #83, Zhuhai Street, Jiaonan, Shandong Province, 266400, China.